MULTICAST FLOW CONTROL IN WIDE-AREA NETWORKS

by

Xi Zhang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2002

Doctoral Committee:
Professor Kang G. Shin, Chair
Professor Semyon M. Meerkov
Professor Robert L. Smith
Professor Wayne E. Stark

UMI Number: 3042206

UMI®

UMI Microform 3042206

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

To my parents

ACKNOWLEDGEMENTS

My deepest gratitude goes to my advisor Professor Kang G. Shin for his constant guidance, encouragement, and support throughout the course of this work. He has always been eager to discuss our research problems and has made many inspiring comments and insightful suggestions without which this work would not have been completed. Moreover, in spite of the tremendous demand on his time, he has always been concerned about my academic progress and personal life. I would also like to express my great appreciation to Professor Semyon M. Meerkov, Professor Roberts L. Smith, and Professors Wayne E. Stark, for serving on this doctoral committee, and for their constructive suggestions.

I have greatly benefited from my discussions with several previous and present members of Real-Time Computing Laboratory, especially, Jennifer Rexford, Sunghyun Choi, Tarek Abdelzaher, Khawar Zuberi, Hengming Zou, Lei Zhou, Ashish Mehra, Anees Shaikh, Padmanabhan (Babu) Pillai, Sung-Whan Moon, Daji Qiao, Chun-Ting Chou, Haining Wang, and Zonghua Gu.

I would also gratefully acknowledge the Department of EECS, the US Office of Naval Research, and National Science Foundation for providing financial support during the course of my graduate study and supporting attendance of international conferences. Thanks to Wee Tech Ng for helping me with the NetSim simulation debugging and numerical-solution programming problems. Thanks also to Huayun Chen, Chuanguo Wang, and all my badminton and table-tennis club fellows, which made my stay in Ann Arbor more enjoyable.

A special thank goes to my former advisor at Lehigh University, PA, the late Professor Richard Denton, who invited me to the USA from Australia, his wife Mrs. Jeanette Denton, and their eleven children and families, who have been taking care of me as my second parents, brothers, and sisters, at my second home in the States — Bethlehem, PA. Finally, I deeply thank my parents and little sister for their constant and never-ending love, crossing the Pacific Ocean via phone calls, emails, and letters.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Multicast Networking and Flow Control

Multicast provides an efficient way of simultaneously disseminating data or information from one source to multiple receivers. Instead of sending a separate copy of the data to each individual receiver using multiple unicasts, the source just sends a single copy once to all the receivers in the multicast group. Conceptually, the underlying multicast-network delivery system forms a *multicast tree* connecting the source and all the receivers, with the sender as the root and the receivers as the leaf nodes. Data or information generated by the sender flows through the multicast tree, traversing each link of the multicast tree exactly once. As a result, multicast offers high efficiency in utilizing network resources and has a wide spectrum of applications, such as software distribution, multimedia streaming, and distance learning/collaboration. Like in unicast, flow control also plays a crucial role in multicast over the best-effort networks, such as the Internet. The purpose of flow control is to minimize the traffic congestion while maximizing the efficiency in network-resource utilization. As a classic and popular research area in the field of networking, the flow-control theory has evolved over the last two or three decades. However, multicast brings out many new challenges in flow control that were not encountered in unicast, and multicast flow control is still in its infancy. The main goal of this dissertation is to develop protocols and modeling techniques to solve the new flow-control problems associated with multicast in wide-area networks.

Different multicast flow-control protocols target at different multicast applications, and also differ in their implementations. Based on different control methods, application objectives, and network structures, multicast flow control can be classified into the following several major categories.

Open-loop vs. Closed-loop. The open-loop multicast flow control is typically used for real-time multimedia streaming applications, such as teleconferencing, where multicast flow control is mainly used for admission control to ensure the admitted multicast users to receive guaranteed QoS (Quality-of-Service). Real-time multicast flows can tolerate a certain level of losses, but are sensitive to large delay or delay jitter, making the closed-loop scheme unsuitable. On the other hand, the closed-loop multicast flow control is essential for data dissemination over the best-effort networks where multicast flow control dynamically adapts the source rate to the variation of the available bandwidth in the network/receivers. While data multicast flows usually do not have to guarantee strict delay-bounds, they must be delivered losslessly, thus requiring closed-loop flow control.

Rate-Based vs. Window-Based. There are mainly two types of multicast flow-control schemes: window-based (e.g., TCP [1]) and rate-based (see [2]). The window-based scheme dynamically adjusts the upper-bound of the number of packets that the transmitter may send without receiving an acknowledgment from the receiver. In the rate-based scheme, the transmitter regulates its sending rate based on network-congestion feedback. The window-based scheme is cost-effective as it does not require any fine-grain rate-control timer, and the window size automatically limits the load a source can impose on the network. However, the window-based scheme also introduces its own problems, including lack of bandwidth guarantee, vulnerability to packet losses and RTT (RoundTrip Time) variation, and complication of error-control mechanism.

Explicit vs. Implicit Feedback. Depending on whether routers perform Active Queue Management (AQM) or not, closed-loop multicast flow control can either use packet drop or duplicate ACKs (using a simple Drop-Tail router) to *imply* the network congestion, or install an AQM mechanism in each router, such as RED (Random Early Detection) and REM (Random Early Marking) or ECN (Explicit Congestion Indication), which explicitly detects and sends the congestion signals to the multicast source. While the implicit feedback minimizes the router complexity, it has two major weaknesses: (1) it cannot distinguish the drops due to congestion from those due to link failures (e.g., due to the noisy wireless links in the multicast tree), and (2) the flow-control scheme drops packets on its own, triggering unnecessary, but expensive, retransmissions. In contrast, explicit feedback can not only avoid the above two problems, but also minimize drops/retransmissions with early congestion detection at the expense of extra router complexity.

Binary vs. M-ary Feedback. A feedback signal can employ one bit where the traffic source makes every flow-control decision based on only a single bit feedback, thus called *binary feedback*, such as TCP's drop-ACK, RED's ECN-bit, ABR's CI-bit, etc., or *M-ary feedback* where each flow-control decision at source is derived from multiple-bit feedback, e.g., Explicit-Rate feedback in ATM networks. While binary feedback minimizes the feedback signaling overhead, it suffers from less dynamic stability and low bandwidth-utilization efficiency, because it only implements coarse-grain flow control. In contrast, M-ary feedback can offer much higher flow-control performance because it applies fine-grained flow-control, accurately adapting the source rate to the actual available bandwidth. However, M-ary feedback is more expensive in both router implementation and bandwidth consumption. This problem becomes even severer in multicast because multicast incurs a much higher volume of flow-control feedback signaling traffic when the number of multicast-tree branches is large.

Deterministic vs. Random Marking. For AQM-equipped routers, the packet ECN-bit can be marked either deterministically as long as the aggregate queue length reaches a predetermined threshold, such as the CI-bit used in ABR, or randomly with a probability proportional to the congestion level measured at the bottlenecked routers, like in RED gateways. The random marking outperforms the deterministic marking for the following four reasons. First, the marking probability of a multicast connection is proportional to its actual bandwidth share so that packets of ill-behaved connections are more likely to get marked. Second, random marking does not have any bias against bursty multicast sources, because a packet is marked based on the average of the aggregate queue length, allowing small bursts to go unharmed and marking every packet only during sustained overloads. Third, random marking can avoid the "global synchronization" problem of deterministic marking that results from many connections reducing their rate at the same time. Finally, as will be shown in Chapter 6, random marking can virtually achieve fine-grain M-ary feedback multicast flow-control performance while only using binary feedback by fusing a sequence of random marks. However, these benefits of random marking are achieved at the cost of the increased complexity of random marking multicast routers.

As is clear from the above discussion, there is no single flow-control scheme which can perfectly satisfy all the multicast flow-control requirements. Each scheme has its own strengths and weaknesses, depending on the application objectives and the performance metrics used. The goal of this dissertation is to make the optimal trade-off among different flow-control schemes by carefully tailoring them to develop new multicast flow-control protocols, which can best solve the new flow-control problems associated with multicast.

1.2 Main Contributions

From the flow-control theory viewpoint, a flow-control scheme or protocol typically consists of two fundamental components: (1) *rate control*— adapting the source rate (or window size) to the dynamic variation of available network bandwidth; and (2) *flow-control signal-*

ing — delivering the flow-control information related to both congestion and rate-control between the source and network/receivers. We address the new flow-control problems associated with multicast by considering these two components as follows.

Multicast rate-control algorithms: The multicast-tree bottleneck round-trip time (RTT) varies when the bottleneck changes from one path to another, which has a significant impact on the multicast flow-control performance [2–7]. To make the multicast flow control scalable to multicast RTT variations, we develop a binary-feedback-based rate-control scheme [2, 5, 6]. At the heart of the proposed scheme is an optimal second-order rate control algorithm, called the *α-control* [8], which adapts the rate ramp-up speed to the variation in RM-cell RTT resulting from dynamic "drift" of the bottleneck in a multicast tree. Applying two-dimensional rate control, the proposed scheme not only makes the rate process converge to the available bandwidth of the connection's most congested link, but also confines the buffer occupancy to a target regime bounded by a finite buffer capacity. Using the fluid analysis, we model the proposed scheme and analyze the system dynamics for multicast ABR traffic. The analytical results show that the proposed scheme is stable and efficient in terms of convergence of source-rate and queue-size to a small neighborhood of the designated operating point. The simulation results verify the analytical findings.

In contrast to the binary feedback based multicast flow control, we also develop a *virtual M-ary* (VMARY) feedback-based multicast flow-control scheme [9, 10], which can achieve a *fine-grained* rate control while keeping the feedback signaling traffic as low as the case of binary feedback. Using the duality theory, we first model the multicast rate control as a distributed optimization problem with a structure separable in both aggregate utilities and constraints. The global optimization objective is to maximize the aggregate utility of all source rates subject to every link's capacity constraint in the multicast tree. We then achieve the optimization by developing a distributed gradient projection algorithm and a random marking based congestion feedback mechanism. The key of the feedback mechanism is the *feedback fusion rule* implemented at each branch router, which aggregates the feedback ECN-bit sequences from all connected downstream branches, and derives a single aggregate ECN-bit sequence. When the aggregate ECN-bit sequence eventually reaches the source, the marking probability of the most-congested path is derived as the M-ary congestion-level feedback information, which is then used to control the next optimization iteration. The feedback fusion rule is easy to implement and proved to be optimal in terms of maximizing the bandwidth utilization and adaptiveness. We model the proposed scheme and compare its performance with the binary-feedback scheme through both analysis and simulation.

Multicast flow-control signaling protocols: For multicast flow-control signaling, there are two new major problems. The first problem is *scalability* — simultaneous arrival of feedback signals from all branches can cause *feedback implosion* [11–13]. Hence, all feedback signals need to be consolidated at all branch points, and then one consolidated feedback is sent to its upstream node. The second problem is *feedback synchronization* — different downstream branches' feedbacks may arrive at the branch point at significantly different times, and the unsynchronized feedback consolidation may mislead the source-rate controller, causing the *consolidation noise* problem [12, 14, 15]. To solve the above two problems with multicast flow-control signaling for ATM ABR services, we propose the *Soft Synchronization Protocol* (SSP) [3] which consolidates the feedback RM cells at each branch point that are not necessarily responses to the same forward RM cell in each synchronization cycle. Through the fluid analysis and simulations, we show that the proposed SSP not only scales well with multicast-tree's height and path lengths [11] while providing efficient feedback synchronization, but also simplifies the implementation of detection and removal of non-responsive branches.

Most previous research on multicast signaling has focused on the algorithm design and implementation. However, the delay properties of these algorithms, despite their vital impact on multicast flow control, are neither well understood nor thoroughly studied. To remedy this deficiency, we develop a balanced and unbalanced binary-tree delay models to study the delay performance of a class of feedback-synchronization signaling protocols, including our SSP and the widely-known hop-by-hop (HBH) algorithm, for multicast ATM ABR flow control. The deterministic binary-tree model is then used to derive a set of expressions for calculating each path's RTT in a given multicast tree. To capture the statistical characteristics of multicast signaling delay when the multicast-tree bottleneck shifts among the multicast-tree paths, we further develop a statistical model to characterize the delay properties for RED- and REM-based multicast flow control, where the random markings at different links are independent. Applying the binary-tree and statistical multicast-signaling delay models, we derive the probability distributions for any path to be the multicast-tree bottleneck and the first and second moments of multicast-signaling delay across the entire multicast. The thus-obtained numerical results also statistically show that SSP outperforms HBH in terms of both the means and variances of the multicast signaling delay. We also conduct extensive simulations, which all verify the analytical findings based on the statistical model, thus confirming the accuracy of the statistical model.

Finally, we consider the general case in which the congestion markings at different links are *dependent*. Including congestion-marking dependencies in the analysis is usually much harder than that under the independence assumption. However, the analysis without assuming independent markings can capture the statistical characteristics more accurately for many practical cases. Specifically, we develop a Markov-chain model defined by the link-marking state on each path in a multicast tree [4, 5]. The Markov chain can not only characterize link-marking dependencies, but also yield a tractable analytical model. We also develop a *Markov-chain dependency-degree model*, which can quantify and evaluate all possible Markov-chain dependency degrees without any prior knowledge of the actual dependency degree. Using the Markov-chain and Markov-chain dependency-degree models, we derive the general expressions for the probability distribution of each path being the multicast bottleneck. Also derived are the closed-form expressions for the first and second moments of multicast signaling delays. The modeling accuracy and analytical findings have been confirmed by simulations. The proposed Markov chain is also shown to asymptotically reach an equilibrium, and its limiting state distribution converges to the marginal link-marking probabilities. We also show that the developed Markov-chain is ergodic if it is irreducible, which is practically useful because it enables us to evaluate the various statistical averages through the sample averages. Applying the developed Markov-chain and Markov-chain dependency-degree models, we also analyze and contrast the delay scalability of SSP and HBH signaling protocols, and the numerical analysis show the superiority of SSP to HBH in terms of multicast signaling delay under dependent link-markings. Again, the obtained analytical findings are all confirmed by simulations.

1.3 Outline of the Dissertation

This dissertation is organized as follows. In Chapter 2, we propose the second-order rate-control based flow-control scheme for multicast[1] ABR service in ATM networks, which can adapt the multicast flow control to multicast RTT variations. To overcome the feedback implosion and feedback synchronization problems, in Chapter 3 we propose the *Soft Synchronization Protocol* (SSP) which consolidates the feedback RM cells at each branch point that are not necessarily responses to the same forward RM cell in each synchronization cycle. Also developed is a binary-tree-based deterministic multicast signaling delay model which generates a set of equations to calculate the RTT for each path in the given multicast tree. In Chapter 4, we develop a *statistical* binary-tree models to study the delay performance of a class of feedback-synchronization signaling algorithms for multicast ATM ABR flow control.In Chapter 5, considering the general case where the congestion markings at different links are *dependent*, we develop a Markov-chain and a Markov-chain

[1]Strictly speaking, multicast includes point-to-multipoint, multipoint-to-point, and multipoint-to-multipoint transmissions. However, for the convenience of presentation, in this dissertation we use the narrow-sense definition for multicast which stands for the point-to-multipoint transmission.

dependency-degree model which are used to derive probability density functions for a path to become the most congested and obtain the first and second moments of multicast signaling delay. In Chapter 6, we develop a *virtual M-ary* (VMARY) feedback optimization multicast flow-control scheme, which can achieve a *fine-grained* rate control while only using the binary feedback. We model the proposed scheme and compare its performance with the binary-feedback scheme using both analysis and simulation. This dissertation concludes with Chapter 7, summarizing the main contributions of this dissertation and discussing future directions.

CHAPTER 2

SCALABLE FLOW CONTROL FOR MULTICAST ABR SERVICES IN ATM NETWORKS

2.1 Introduction

The ABR flow-control algorithm has two major functions: determining the bottleneck link bandwidth, and adjusting the source transmission rate to match the bottleneck link bandwidth and buffer capacity. In a multicast ABR connection, determining the bottleneck link bandwidth is a daunting task. The first generation of multicast ABR algorithms [16–19] employ a simple hop-by-hop feedback mechanism for this purpose. In these algorithms, feedback RM (Resource Management) cells from downstream nodes are consolidated at branch points. On receipt of a forward RM cell, the consolidated feedback is propagated upwards by a single hop. While hop-by-hop feedback is very simple, it does not scale well because the RM-cell RTT is proportional to the height of the multicast tree. Moreover, unless the feedback RM cells from the downstream nodes are *synchronized* at each branch point, the source may be misled by the incomplete feedback information, which can cause the *consolidation noise* problem [20].

In order to reduce the RM-cell RTT and eliminate consolidation noise, the authors of [15, 20] proposed feedback synchronization at each branch point by accumulating feedback from *all* downstream branches. The main problem with this scheme is its slow transient response since the feedback from the congested branch may have to needlessly wait for the feedback from "longer" paths, which may not be congested at all. Delayed congestion feedback can cause excessive queue build-up and cell loss at the bottleneck link. The authors of [21] proposed an improved consolidation algorithm to speed up the transient response by sending the fast overload-congestion feedback without waiting for all branches' feedback during the transient phase.

One of the critical deficiencies of the schemes described above is that they do not detect and remove non-responsive branches from the feedback synchronization process. One or more non-responsive branches may detrimentally impact end-to-end performance by providing either stale congestion information, or by stalling the entire multicast connection. We propose a *Soft-Synchronization Protocol* (SSP) which derives a consolidated RM cell at each branch point from feedback RM cells of different downstream nodes that are not necessarily responses to the same forward RM cell in each synchronization cycle. The proposed SSP not only scales well with multicast-tree's height and path lengths [11] while providing efficient feedback synchronization, but also simplifies the implementation of detection and removal of non-responsive branches. A scheme similar in spirit but different in terms of implementation has been proposed independently in [20] and [15].

As clear from the above discussion, the problem of determining the bottleneck link bandwidth in a multicast ABR connection has been addressed by many researchers. Unfortunately, little attention has been paid to the problem on how to adjust the transmission rate to match the bottleneck bandwidth and buffer capacity in the multicast context. All of the schemes proposed in the literature retrofit the transmission control mechanism used for unicast ABR connections to multicast connections. Consequently, they have overlooked an important but subtle problem that is unique to multicast ABR connections. Unlike in unicast, in a multicast connection the bottleneck may shift from one path to another within the multicast tree. As a result, the RM-cell RTT in the bottleneck path may vary significantly. Since the RTT plays a critical role in determining the effectiveness of any feedback flow-control scheme, it is important to identify and handle such dynamic drifts of the bottleneck. Failure to adapt with RM-cell RTT variations may either lead to large queue build-ups at the bottleneck or slow transient response.

A key component of the scheme proposed in this chapter is an optimal second-order rate control algorithm, called the *α-control*, designed to cope with RM-cell RTT *variations*. Specifically, the proposed rate control scheme not only regulates the traffic source rate based on the congestion feedback, but also adjusts the rate-gain parameter α, which is the speed of rate increase. As will be discussed later, the maximum queue-size is an increasing function of both the RM-cell RTT and the rate-gain parameter α, and the α-control can make the flow-control performance dynamically adaptive to RM-cell RTT variations. The formal introduction and fundamental principle of the α-control will be detailed in Section 2.4.2. Using the fluid analysis, we model the α-control with the binary-congestion feedback, and study the system dynamics in the scenarios of both persistent and on-off ABR traffic sources. We develop an optimal control condition, under which the α-control guarantees the monotonic convergence of system state to the optimal regime from an arbitrary initial value. The analytical results show that the proposed scheme is efficient and stable in that both the source rate and bottleneck queue length rapidly converge to a small neighborhood of the designated operating point. The α-control is also shown to adapt well to RM-cell RTT variations in terms of buffer requirements and fairness.[1] The dynamic performance of the proposed scheme is evaluated by both modeling analysis and simulation experiments, and the simulation results verify the analytical results. To analyze the performance of the proposed scheme in more general network scenarios, we also conducted extensive simulations for the case of *concurrent* multiple multicast connections where the number, location, and bandwidth of bottlenecks vary dynamically. The simulation results demonstrate the superiority of the proposed scheme to the other schemes in dealing with

[1] The definition of fairness used throughout this chapter is adopted from [22] where the fairness is achieved when all connections receive an equal share/allocation of the network resources (bandwidth or buffer capacities). This differs from the max-min fairness, which deals with more general cases where some connections' demand is smaller than an equal share/allocation of the network resources.

the variations of RM-cell RTT and link bandwidth, achieving fairness in both buffer and bandwidth occupancies, and improving average throughput.

This chapter is organized as follows. Section 2.2 describes the proposed scheme. Section 2.3 establishes the flow-control system model. Section 2.4 justifies the necessity and feasibility of the α-control, presents the α-control algorithm, and investigates its properties. Section 2.5 derives analytical expressions for both transient and equilibrium states, evaluates the scheme's performance for the single-connection case, derives the greatest lower bound of target buffer occupancy, conducts loss control analysis, and compare the analysis and simulation results. Section 2.6 investigates the convergence to fairness and aggregate efficiency of the proposed α-control among the multiple concurrent multicast connections, analyzes the flow-control dynamic performance of concurrent multiple multicast-connections by the fluid analysis, and compares the proposed scheme with the other existing schemes through simulations. The chapter concludes with Section 2.7.

2.2 The Proposed Scheme

Based on the ABR flow-control framework in [23], we use RM cells to convey network-congestion information. A forward RM cell is sent by the root (source) node periodically or once every N_{rm} data-cells, and each receiver node replies by returning to the source a feedback RM cell with CI (Congestion Indication) and ER (Explicit Rate) information. We redefine the RM-cell format by adding information on the rate-gain parameter (second-order) control in the standard RM cell to deal with RM-cell RTT variations. In particular, two new one-bit fields, BCI (Buffer Congestion Indication) and NMQ (New Maximum Queue), are defined. Our scheme distinguishes the following two types of congestion:

Bandwidth Congestion: If the queue length $Q(t)$ at a switch becomes larger than a predetermined threshold Q_h, then the switch sets the local CI (Congestion Indication) bit to 1.

```
On receipt of an RM cell:
  if (LCI=1 ∧ CI=0) {                              ! Buf-congest control trigger condition
    if (BCI=1) {AIR := q × AIR};                   ! AIR reduced multiplicatively
    elseif (BCI=0 ∧ LBCI=0) {AIR := p + AIR};      ! Increase AIR additively
    elseif (BCI=0 ∧ LBCI=1) {AIR := AIR/q};        ! BCI toggles from 1 to 0, around AIR
    MDF := e^(-AIR/BW_EST);                        ! MDF updating
    LNMQ := 1;        LBCI := BCI; };              ! Start a new measurement cycle
  if (CI=0) {ACR := ACR + AIR};                    ! Increase cell rate additively
  else {ACR := ACR × MDF};                         ! Decrease cell rate multiplicatively
  LCI := CI;                                       ! Save CI and BCI bits for α-control.
```

Figure 2.1: The pseudo-code for Source End System (SES).

***Buffer Congestion*:** If the maximum queue length Q_{max} at a switch exceeds the target buffer occupancy Q_{goal}, where $2Q_h < Q_{goal} < C_{max}$ (see Theorem 2.5.2) and C_{max} is the buffer capacity, then the switch sets the local BCI to 1.

Notice that the buffer congestion represents a severer congestion condition than the bandwidth congestion, and thus, the buffer congestion always occurs after the bandwidth congestion already exists. As will be elaborated on later, the bandwidth congestion control deals with the link bandwidth constraint while the buffer congestion is targeted at the buffer occupancy control.

2.2.1 The Source Algorithm

A pseudocode for the source control algorithm is presented in Figure 2.1. Upon receiving a feedback RM cell, the source must first check if it is time to exercise the buffer-congestion control (the α-control). The buffer-congestion control is triggered when the source detects a transition from a rate-decrease phase to a rate-increase phase, that is, when LCI (local congestion indicator) is equal to 1, and the CI field in the RM cell received is set to 0. The rate-gain parameter is adjusted according to the current value of the local BCI ($LBCI$) and the BCI field in the RM cell just received. There are three different variations: (i) if BCI is set to 1 in the RM cell received, the rate-gain parameter AIR (Additive Increase Rate) is decreased multiplicatively by a factor of q ($0 < q < 1$); (ii) if both $LBCI$ and BCI are set to

0, the rate-gain parameter AIR is increased additively by a step of size $p > 0$; (iii) if $LBCI = 1$ and $BCI = 0$, AIR is increased multiplicatively by the same factor of q. For all of these three cases, the rate-decrease parameter MDF (Multiplicative Decrease Factor) is adjusted according to the estimated bottleneck bandwidth BW_EST. Then, the local NMQ bit is marked and the BCI-bit in the RM cell received is saved in $LBCI$ for the next α-control cycle. The source always exercises the cell-rate (first-order) control whenever an RM cell is received. Using the same, or updated, rate-parameters, the source additively increases, or multiplicatively decreases, its ACR (Allowed Cell Rate) according to the CI-bit in the RM cell received. Based on the source algorithm, Figure 2.5 in Section 2.5 demonstrates the equilibrium dynamics of the source rate $R(t)$ and the bottleneck queue length $Q(t)$, using the fluid functions to be discussed in Section 2.3. Driven by feedback CI-bit, $R(t)$ fluctuates around the bottleneck bandwidth, but alternates between two different ramp-up speeds determined by the feedback BCI-bit. As a result, the maximum queue length $Q_{max}^{(n)}$ at the bottleneck is confined to the designated operating regime around the target buffer occupancy Q_{goal}.

2.2.2 The Switch Algorithm

At the center of switch control algorithm is a pair of connection-update vectors: (I) *conn_patt_vec*, the connection pattern vector where *conn_patt_vec*$(i) = 0$ (1) indicates the i-th output port of the switch is (not) a downstream branch of the multicast connection. Thus, *conn_patt_vec*$(i) = 0$ (1) implies that a data copy should (not) be sent to the i-th downstream branch and a feedback RM cell is (not) expected from the i-th downstream branch;[2] (II) *resp_branch_vec*, the responsive branch vector is initialized to $\underline{0}$ and reset to $\underline{0}$ whenever a consolidated RM cell is sent upward from the switch. *resp_branch_vec*(i) is set to 1 if a feedback RM cell is received from the i-th downstream branch. The connection pattern of *conn_patt_vec* is updated by *resp_branch_vec* each time when the non-responsive branch is detected or a new connection request is received from a downstream branch.

[2] Note that the negative logic is used for convenience of implementation.

```
On receipt of a DATA cell:
  multicast DATA cell based on conn_patt_vec;       ! multicast data cell to connected branches
  if (data_qu > Q_h)  {CI :=1;}                     ! 1) Bandwidth congestion control
  if (data_qu > Q_max)  {Q_max :=data_qu;}          ! 2) update Q_max
  if (Q_max > Q_goal)  {BCI :=1;}                   ! 3) buffer congestion control
  else {BCI :=0;}                                   ! 1), 2), 3) applied to all connctd brches
On receipt of a feedback RM cell  from i-th downstream branch:
  if (conn_patt_vec(i) ≠ 1) {                       ! only process connected branch
    resp_branch_vec(i) := 1;                        ! mark connected/responsive branch
    MCI:= MCI ∨ CI;                                 ! bandwidth-congestion indicator process
    MBCI:= MBCI ∨ BCI;                              ! buffer-congestion indicator processing
    MER:= min{MER, ER};                             ! ER information processing
    if (conn_patt_vec ⊖ resp_branch_vec =1) {       ! soft-synchronization
     send RM cell (dir := backward,  ER:= min_{resp-branches} MER, CI:= ∪_{resp-branches} MCI,
          BCI := ∪_{resp-branches} MBCI);           ! send fully consolidated RM cell upwards
     resp_branch_vec:= 0;                           ! reset responsive branch vector
     MCI := 0;  MER := ER;                          ! reset congest control variables of RM cell
     no_resp_timer := N_nrt;}}                      ! reset non-responsive timer
On receipt of a forward RM cell:
  multicast RM cell based on conn_patt_vec;         ! multicast RM cell
  if (NMQ=1)  {MBCI:=0;  Q_max := 0;}               ! start a new measurement cycle
  no_resp_timer := no_resp_timer - 1;               ! no-responsive branch checking
  if (no_resp_timer = 0){                           ! there is a no-responsive branch
    conn_patt_vec := resp_branch_vec ⊖ 1;           ! update connection pattern vector
    if (resp_branch_vec ≠ 0){                       ! there is at least one responsive branch
      send RM cell (dir := backward,  ER:= min_{resp-branches} MER, CI = ∪_{resp-branches} MCI,
        BCI:=∪_{resp-branches} MBCI);               ! send partial consolidated RM cell up
      resp_branch_vec := 0;                         ! reset the responsive branch vector
      MCI := 0;  MER := ER;                         ! reset congest control variables of RM cell
      no_resp_timer := N_nrt;}}                     ! reset the non-responsive timer
On receipt of connect/dis-connect request from j-th downstream branch:
  conn_patt_vec(j) := 0;                            ! add/reactivate a branch in multicast tree
  conn_patt_vec(j) := 1;                            ! disconnect a branch in multicast tree.
```

Figure 2.2: The pseudo-code for Intermediate Switch System (ISS).

A simplified pseudocode of the switch control algorithm is given in Figure 2.2. Upon receiving a data cell, the switch multicasts the data cell to its output ports specified by *conn_patt_vec*, if the corresponding output links are available, else enqueues the data cell in its branch's queue. Mark the branch's CI ($EFCI$) if queue length $Q(t) > Q_h$. Update Q_{max} for the α-control (to be discussed in Section 2.4.1) if the branch's new $Q(t)$ exceeds the old Q_{max}. $BCI := 1$ if its updated $Q_{max} \geq Q_{goal}$, the target buffer occupancy.

On receipt of a feedback RM cell returned from either one of the receivers or a connected downstream branch, the switch first marks its corresponding bit in *resp_branch_vec* and then performs RM-cell consolidation operations, using OR rule. If the modulo-2 addition (the soft-sychcronization operation of SSP), $conn\_patt\_vec \oplus resp\_branch\_vec = \underline{1}$, an all 1's vector, indicating all feedback RM cells synchronized, then a fully-consolidated feedback RM cell is generated and sent upward. But, if the modulo-2 addition is not equal to $\underline{1}$, the switch needs to await other feedback RM-cells for synchronization. Since the switch control algorithm does not require that a consolidated RM-cell be derived from only those feedback RM-cells corresponding to the same forward RM-cell, the feedback RM-cell consolidation is "softly-synchronized".

Upon receiving a forward RM-cell, the switch first multicasts it to all the connected branches specified by $conn\_patt\_vec$. Then, reset $Q_{max} := 0$ and the buffer congestion indicator $MBCI := 0$ if an NMQ request is received. The non-responsive timer no_resp_timer is initialized to a threshold N_{nrt} and reset to N_{nrt} whenever a consolidated RM-cell is sent upward. The predetermined timeout value N_{nrt} for non-responsiveness is determined by such factors as the difference between the maximum and minimum RM-cell RTTs. We use the forward RM-cell arrival time as a natural clock for detecting/removing non-responsive branches (such that it will still work even in the presence of faults in the downstream branches). Each time a switch receives a forward RM-cell, the multicast connection's no_resp_timer is decreased by one. If $no\_resp\_timer = 0$ (timeout) and $resp\_branch\_vec \neq \underline{0}$ (i.e., there is at least one downstream responsive branch), then the switch will stop awaiting arrival of feedback RM-cells and immediately generate a partially-consolidated RM-cell, and send it upward. Whenever $no\_resp\_timer = 0$ is detected, at least one non-responsive downstream branch is detected and will be removed by the simple operation: $conn\_patt\_vec := resp\_branch\_vec \oplus \underline{1}$.

Therefore, a downstream branch which has not sent any feedback RM-cell for N_{nrt} forward RM-cell time units will be removed from the multicast tree. On the other hand, a downstream node can join the multicast connection, at run-time, by submitting a join-in request to its immediate upstream branch-switch. So, our algorithm supports the dynamic reconfiguration of a multicast tree.

2.2.3 Multicast Flow-Control Signaling and Scalability

The multicast flow-control algorithms proposed above consists of two basic components: flow-control signaling and rate control. These two components are conceptually separate from a flow-control theory viewpoint, even though they are blended together in the proposed algorithms. We first consider flow-control signaling scalability in this section, and will then focus on the rate control in the rest of the chapter.

As clear from the proposed algorithms, the flow-control signaling relies on RM cells, which deliver the rate-control and congestion information between the source-rate controller and the network/receivers. Signaling for multicast flow control imposes two new challenges: *scalability* and *feedback-synchronization*. These two problems are closely related in the signaling protocol for multicast flow control. First, if multicast flow-control signaling is implemented naively such that each receiver sends its own feedback RM cell individually all the way back to the source, the flow-control traffic due to feedback RM cells will result in "feedback explosion" — not only at the source but at all branch nodes of the multicast session. Hence, it is important for each branch point to consolidate the congestion-information feedback from its downstream nodes and send only the consolidated feedback to its upstream node. Second, we need a feedback-synchronization signaling algorithm to synchronize the feedback-consolidation at each branch point, because different downstream branches' feedback may arrive at the branch point at significantly different times, and the unsynchronized feedback-consolidation may mislead the source-rate control decisions, causing the *consolidation noise* problem [15,20].

To solve the above two problems with multicast flow-control signaling, we propose the *Soft Synchronization Protocol* (SSP) which consolidates the feedback RM cells at each branch point that are not necessarily responses to the same forward RM cell in each synchronization cycle. The algorithm of SSP is detailed in Figure 2.2. Each multicast switch achieves once of feedback synchronization and sends a consolidated RM cell to the upstream node as long as it receives at least one feedback RM cell from each of all the connected downstream branches since the last sychronization cycle. Since SSP allows the feedback RM cells corrsponding to different forward RM cell to be consoliated (soft-synchronization) at each branch point, the effective RM-cell RTT can be as small as the shortest path RTT, and virtually independent of the multicast-tree height as compared to the widely-known hop-by-hop feedback-synchronization scheme [16]. Moreover, SSP also ensures that the ratio of feedback RM cells to forward RM cells is no larger than 1 at each link of the multicast tree. Thus, SSP scales well with the multicast session size in terms of both RM-cell RTT delay and feedback signaling traffic. The scalability of SSP in terms of RM-cell RTT delay is quantitatively studied in [11].

2.3 The System Model

The proposed scheme can support both (1) CI-based rate control with a binary congestion feedback (CI-bit); and (2) ER-based rate-control with an explicit-rate feedback (ER-value). The CI-based scheme is more suitable for LANs because of its minimum multicast signaling cost and lowest implementation complexity. As compared to the CI-based scheme, the ER-based scheme is more responsive to network congestion and can better serve WAN environments where the bandwidth-delay product is large. However, the ER-based scheme is much more expensive to implement than the CI-based scheme. In this chapter, we will focus only on the CI-based scheme. The rate-control algorithm and the α-control to be discussed later will be only for the CI-based scheme, not for the ER-based scheme. We model the CI-based flow-control system by the first-order fluid analysis [7,24–26], which uses the continuous-time functions $R(t)$ and $Q(t)$ as the fluid function of the source rate and bottleneck queue length, respectively. We also assume the existence of only a single bottleneck[3] on each path at a time with queue length equal to $Q(t)$ and a "persistent"

[3]This is not a restriction, because the bottleneck is defined as the most congested link or switch on a path.

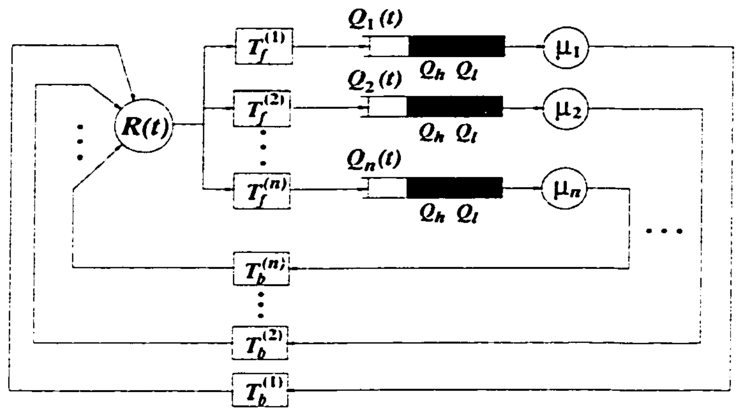


Figure 2.3: The system model for a multicast connection.

source with $ACR = R(t)$ for each multicast connection.

2.3.1 System Description

As shown in Figure 2.3, a multicast-connection model consists of n paths with RM-cell RTTs $\tau_1, \tau_2, \cdots, \tau_n$, and bottleneck link bandwidths $\mu_1, \mu_2, \cdots, \mu_n$. There is only a single bottleneck on each path and its location may change with time. $T_f^{(i)}$ represents the "forward" delay from the source to the bottleneck, and $T_b^{(i)}$ the "backward" delay from the bottleneck to the source via the receiver of the i-th path. Clearly, $T_b^{(i)} = \tau_i - T_f^{(i)}$. Notice that because ABR flow control in ATM network typically employs a special control channel (an out-of-rate channel) [23, 27] to convey RM cells to avoid queueing delay of flow-control signaling messages, RTT in the proposed model, $\tau_i = T_b^{(i)} + T_f^{(i)}$, for $i = 1, 2, \cdots, n$, only considers the propagation delay without including queueing delay. Each path's bottleneck has its own queue length function $Q_i(t), i = 1, 2, \cdots, n$. All paths in a multicast connection "interact" with one another via their "shared" source rate $R(t)$.

We use the synchronous model by assuming that the source sends RM cells periodically with an interval Δ equal to a fraction of RTT. The additive increase and multiplicative decrease of rate control during the n-th rate-update interval can be expressed as:

$$R_n = \begin{cases} R_{n-1} + a; & \text{additively increase, } a = AIR \\ bR_{n-1}; & \text{multiplicatively decrease, } b = MDF \end{cases} \tag{2.1}$$

where $a > 0$ and $0 < b < 1$.

2.3.2 System Control Factors

In unicast ABR service, the source rate is regulated by the feedback from the most congested link/switch which has the minimum *available* bandwidth along the path from source to destination. A natural extension of this strategy to multicast ABR service is to adjust the source rate to the minimum available bandwidth share of the multicast-tree's most congested path that the traffic source has sensed. This is the key feature of ABR service, most suitable for data applications that require lossless transmission. However, the dynamics of multicast ABR flow control is more complicated than those of unicast ABR flow control, because not only the available bandwidth, but also the RTT and congestion threshold can differ from one path to another in a multicast tree. As a result, while the source rate always converges to the available bandwidth of the slowest path *perceived/sensed by the traffic source* (which is not necessarily the currently slowest path in the multicast tree), it is possible that in the transient state the dynamics of source rate is dictated by the feedback via the path with a bandwidth larger than the current minimum available bandwidth across the multicast-tree, depending on the path's RTT and congestion threshold. To explicitly model these features for the multicast flow control, we introduce the following definition.

Definition 2.3.1 *The* **multicast-tree bottleneck path** *(also simply called* **multicast-tree bottleneck***) is the path whose congestion feedback* **currently received at the source** *dictates (or dominates) the source rate-control actions. The* **multicast-tree RM-cell RTT** *is the RM-cell RTT experienced on the multicast-tree bottleneck path.* ■

Remarks on Definition 2.3.1.

R1. The multicast-tree bottleneck path is a *source flow-control oriented* concept/notion because only the congestion-information feedback *currently received by the source* can affect the *current* source flow-control decisions. The current congestion information detected at switches does not affect the source's flow control actions until it reaches the source after a certain delay. So, it is the congestion-information feedback currently received or perceived/sensed by the source, instead of the congestion information currently detected at the switches, that decides which path is the multicast-tree bottleneck at the current moment. Thus, at a given time instant the multicast-tree bottleneck path is ***not*** necessarily always the slowest (with the minimum available bandwidth) path in the multicast tree.

R2. The multicast-tree bottleneck can be formed in one of the following two different phases:

(1) Congested phase: If $CI = 1$ in the currently received consolidated RM-cell at the source and if this $CI = 1$ has resulted from $m \geq 1$ paths with $CI(i) = 1$, then the shortest (with the minimum RTT (τ))[4] of these m paths is the multicast-tree bottleneck. This is because the shortest path among the congested paths *perceived/sensed by the source* determines the RTT of multicast-tree's feedback control loop and the dynamics of the multicast-tree bottleneck as far as the source rate control is concerned;

(2) Non-congested phase: If $CI = 0$ (non-congestion) in the currently received consolidated RM-cell at the source, then the shortest path among all paths, which will cause congestion immediately after this non-congested phase, is the multicast-tree bottleneck. This is also because the shortest congested path *perceived/sensed by the source* determines the RTT of multicast-tree's feedback con-

[4] The information on the other two parameters, available bandwidth (μ_i) and congestion threshold ($Q_h^{(i)}$), has been included in $CI(i) = 1$.

trol loop and the dynamics of the multicast-tree bottleneck as far as the source rate control is concerned;

R3. The multicast-tree bottleneck can change instantaneously as a function of time (even within a rate-control fluctuation cycle), but only at the one of the following two types of transition instants:

(1) When the consolidated RM-cell's CI changes $1 \rightarrow 0$;

(2) When the source-received congestion information $CI(i)$ for the shortest path P_i among all congested paths changes $1 \rightarrow 0$; or the source-received congestion information $CI(i)$ for a non-congested path P_i, which is shorter than all congested paths, changes $0 \rightarrow 1$, while the consolidated RM-cell $CI = 1$ remains unchanged.

R4. From the above remarks **R1** through **R3**, it is clear that the location of the multicast-tree bottleneck path is a function of the available bandwidth (μ_i) in the bottleneck switch on path P_i, the congestion-detection queue threshold ($Q_h^{(i)}$) in the bottleneck switch on P_i, and RTT (τ_i) of path P_i. In addition, it is possible that during the transient state, the multicast-tree bottleneck is not the path that has the minimum available bandwidth in the bottleneck switch across the multicast tree, depending on the path's RTT and congestion threshold.

R5. Since at any given time instant there exists only one shortest path among the congested paths *perceived/sensed by the source* when the congested phase starts, according to remark **R2**, there is only one multicast-tree bottleneck at any given time instant, unless there are more than one path, which have exactly the same RTT (τ_i) on each path and become the congested path at exactly the same time. In that case, albeit not very often in practice, these paths either have exactly the same rate control parameters (μ, Q_h, and τ) or generate feedbacks having the identical effect on the source rate control, and thus, we can arbitrarily choose any one of them as the multicast-tree bottleneck such that the uniqueness of the multicast-tree bottleneck in a multicast tree for any given time instant still holds.

2.3.3 The State Equations for the Multicast-Tree Bottleneck Path

As clear from the above discussion, the multicast-tree bottleneck dictates the source rate-control actions, and thus we can analyze the multicast flow-control system by focusing on its multicast-tree bottleneck's state equations. Let $R(t)$ and $Q(t)$ be the fluid functions of the source rate and the queue length at the current multicast-tree bottleneck defined by Definition 2.3.1, respectively. Then, the multicast-tree bottleneck state is specified by the two state variables, $R(t)$ and $Q(t)$. According to the rate-control algorithms described by Eq. (2.1), the state equations of multicast-tree bottleneck, which is *unique* based on the **Remarks on Definition 2.3.1**, in the continuous-time domain are given by:

Source-rate function:

$$R(t) = \begin{cases} R(t_0) + \alpha(t - t_0); & \text{if } Q(t - T_b) < Q_l \\ \\ R(t_0)e^{-(1-\beta)\frac{(t-t_0)}{\Delta}}; & \text{if } Q(t - T_b) \geq Q_h \end{cases} \tag{2.2}$$

Multicast-tree bottleneck queue function:

$$Q(t) = \int_{t_0}^{t} [R(v - T_f) - \mu] \, dv + Q(t_0), \tag{2.3}$$

where $\alpha = a/\Delta$ and $\beta = 1 + \log b$ (a and b are defined in Eq (2.1)); t is the current observation time of the system states for the current multicast-tree bottleneck path, t_0 is the last observation time of the system states for the current multicast-tree bottleneck path, and t is chosen such that, during the time period of $(t - t_0)$, the **multicast-tree bottleneck path** is fixed and unique and also, during $(t - t_0)$, $R(t)$ is either only in its increasing phase or only in its decreasing phase; $\tau = T_f + T_b$ is the current multicast-tree RM-cell RTT defined by Definition 2.3.1 and RTT, $\tau = T_f + T_b$, in our proposed model only considers the propagation delay without including the queueing delay as detailed in Section 2.3.1; Q_h (Q_l) is the high (low) buffer queue-threshold for the current multicast-tree bottleneck defined by Definition 2.3.1; μ is the available bandwidth of the current multicast-tree bottleneck defined by Definition 2.3.1 (Note that μ is the minimum available bandwidth currently *perceived/sensed* by the source, which is *not* necessarily the true current minimum available bandwidth of the path across the entire multicast tree).

Remark on the system state equations Eqs. (2.2) and (2.3): Fluid analysis is a time-period piece-wise modeling procedure [28]. So, we can use a set of system state equations Eqs. (2.2) and (2.3) of the same form to model the dynamics of the different multicast-tree bottleneck path during the different time period, by replacing the system state variables, such as $Q(t)$, $Q(t - T_b)$, T_b, and T_f (or RTT delay $\tau = T_f + T_b$) for different time periods corresponding to different multicast-tree bottleneck paths. Consequently, the system state variables $Q(t)$, $Q(t-T_b)$, T_b, and T_f (or RTT delay $\tau = T_f+T_b$) given in Eqs. (2.2) and (2.3) are *not* constant because they may be associated with a *different* multicast-tree bottleneck path during a *different* time period of $(t-t_0)$, depending on which path is the multicast-tree bottleneck during that time period of $(t-t_0)$. Even though the multicast-tree bottleneck can change during any time period, the multicast-tree bottleneck path that the traffic source can perceive is *unique* because the queue-length threshold testing: $Q(t - T_b) \geq Q_h$ or $Q(t - T_b) < Q_l$ is only sampled at the time *instants* [5] which are the integer multiples of Δ (where Δ is the RM-cell update time interval). This feature of the proposed multicast flow control algorithm ensures that fluid analysis expressed by Eqs. (3.2) and (3.3) can accurately capture the dynamics of multicast-tree bottleneck path under the proposed multicast flow control algorithm even when the multicast tree bottleneck path changes from one path to another, as long as we take $(t-t_0) < \Delta$ or make $(t-t_0)$ small enough such that the bottleneck path that the traffic source can perceive is always unique[6] during $(t - t_0)$. As a result, the system state equations Eqs. (3.2) and (3.3) characterize the multicast flow-control dynamics

[5] Only at these sampling time instants, the traffic source can perceive/sense the possible change of multicast-tree bottleneck path, and between any two consecutive sampling time instants (separated apart by a time period of Δ, i.e., the RM-cell update time interval) the traffic source does not have a chance to perceive/sense any change of multicast-tree bottleneck path. So, the multicast-tree bottleneck path that the traffic source can perceive remains to be unique (the same) during the time period between any two consecutive sampling time instants.

[6] The uniqueness of the multicast tree bottleneck path, which can be perceived by the traffic source, can be always achieved either by letting $(t - t_0) < \Delta$, or otherwise (if $(t - t_0) > \Delta$) by letting $(t - t_0)$ be small enough such that multicast tree bottleneck path that the traffic source can perceive is unique during $(t - t_0)$.

by modeling the flow-control dynamics of the different multicast-tree bottleneck paths, one path for each time-period of $(t - t_0)$ (piece-wise modeling in terms of time period), as the multicast-tree bottleneck changes from one path during a time-period to another path during the next time-period.

2.4 Adaptation to Variations of Multicast-Tree RM-Cell RTT

In a real network environment, there is always cross-traffic at each link, which may cause the multicast-tree bottleneck path to shift from one path to another. So, the multicast-tree RM-cell RTT fluctuates dynamically between $\tau_{min} \triangleq \min_{1 \le i \le n}\{\tau_i\}$ and $\tau_{max} \triangleq \max_{1 \le i \le n}\{\tau_i\}$. The main and direct impact of RM-cell RTT variations is on the maximum buffer requirement for the multicast-tree bottleneck path.

2.4.1 Maximum Buffer Requirement and Cell-Loss Control

Although SSP makes the RM-cell RTT τ for the proposed scheme much smaller than that for the hop-by-hop scheme, as shown in [11], τ's swing between τ_{min} and τ_{max} is still large enough to make a significant impact on Q_{max}. As discussed in [7], increasing or decreasing $R(t)$ is not effective enough to have the maximum queue length Q_{max} upper-bounded by the maximum buffer capacity C_{max} when the multicast-tree RM-cell RTT τ varies due to drift of the multicast-tree bottleneck. This is because rate-increase/decrease control can only make $R(t)$ fluctuate around the designated bandwidth, but cannot adjust the rate-fluctuation amplitude that determines Q_{max}. So, Q_{max} also depends on the source rate-gain parameter α (to be detailed in Section 2.5). Q_{max} is analytically shown in [7] to increase with both τ and rate-gain parameter $\alpha = \frac{dR(t)}{dt}$ and can be written as a function, $Q_{max}(\alpha, \tau)$, or $Q_{max}(\alpha)$ for a given τ. In reality, the buffer capacity, C_{max}, on the bottleneck path is finite, and hence, to ensure cell-lossless transmission, the condition $Q_{max} \le C_{max}$ must hold. This constraint divides the 2-dimensional (α, τ)-space into two regions as follows.

Definition 2.4.1 *If $C_{max} < \infty$, then the feasible (α, τ)-space, $\Omega \triangleq \{(\alpha, \tau) \mid \alpha > 0, \tau >$*

$0\}$ *is partitioned into two parts:* ***lossless transmission region:*** $\mathcal{F} \triangleq \{(\alpha, \tau) \mid (\alpha, \tau) \in \Omega,$ $Q_{max}(\alpha, \tau) \leq C_{max}\}$ *and* ***lossy transmission region:*** $\mathcal{L} \triangleq \Omega \setminus \mathcal{F}$. ■

The theorem presented below gives an upper bound for the equilibrium-state maximum queue length $Q_{max}(\alpha, \tau)$ as a function of $(\alpha, \tau) \in \Omega$ and Q_h.

Theorem 2.4.1 *Consider a multicast-tree bottleneck characterized by the flow-control parameters* α, τ, *and* Q_h. *If* $(\alpha, \tau) \in \Omega$ *and* $\alpha \left(\frac{\Delta}{1-\beta} \right) = \mu$,[7] *then the maximum queue length is upper-bounded by*

$$Q_{max}(\alpha, \tau) \leq (\tau\sqrt{\alpha} + \sqrt{2Q_h})^2. \tag{2.4}$$

Proof. The proof is given in Appendix A. ■

The upper-bound function of $Q_{max}(\alpha, \tau)$ described in Theorem 2.4.1 provides a closed-form expression that reveals an analytical relationship among the maximum buffer requirement and rate-control parameters. As suggested by Theorem 2.4.1 and also analyzed in [7, 24, 28–30], $Q_{max}(\alpha, \tau)$ is a monotonic increasing function of both α and τ, and thus can be controlled by adjusting α for a given τ. The theorem given below establishes an explicit relationship among α, τ, and Q_h subject to lossless transmission and $C_{max} < \infty$ constraints.

Theorem 2.4.2 *Consider a multicast connection flow-controlled by the proposed scheme with* $Q_h > 0$ *and* $C_{max} < \infty$ *at the multicast-tree bottleneck. If* $C_{max} > 2Q_h$, ***then the*** *following claims hold:*

__**Claim 1:**__ $\mathcal{F} \neq \emptyset$ *and* $\exists\ K > 0$ *such that* $(\alpha, \tau) \in \mathcal{F}\ \ \forall\ (\alpha, \tau) \in \{(\alpha, \tau) \mid \tau\sqrt{\alpha} \leq K, (\alpha, \tau) \in \Omega\}$;

[7] The constraint $\alpha \left(\frac{\Delta}{1-\beta} \right) = \mu$ is set to balance the increasing and decreasing speeds of $R(t)$ [28].

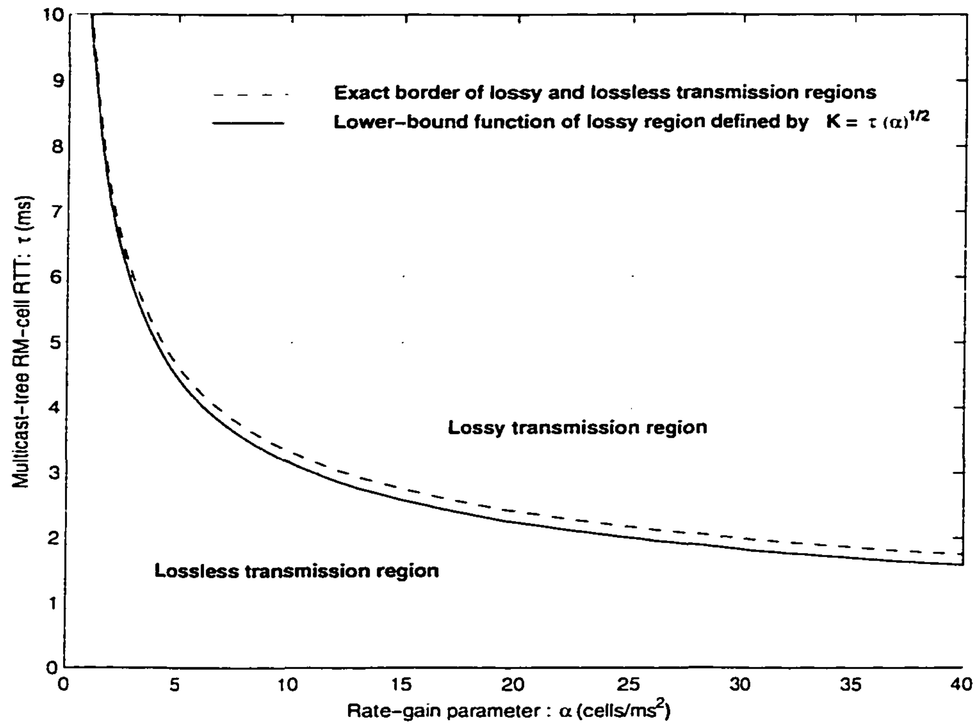


Figure 2.4: Lossy and lossless transmission regions divided by the lower bound of lossy-transmission region.

<u>Claim 2:</u> *$\mathcal{L}$ is lower-bounded by the function $K_\ell = \tau\sqrt{\alpha}$ where $K_\ell = \sqrt{C_{max}} - \sqrt{2Q_h}$ and $(\alpha, \tau) \in \Omega$.*

Proof. The proof is provided in Appendix B. ■

Remarks on Theorem 2.4.2. **(1)** Claim 1 shows that Q_{max} is controllable, and identifies a sufficient condition ($C_{max} > 2Q_h$) for the feasibility of lossless transmission. Moreover, Claim 1 describes the configuration of the lossless-transmission region defined in Ω. **(2)** Claim 2 gives a lower bound of the lossy transmission region $\mathcal{L}$ for given C_{max} and Q_h, which is expressed by a continuous function defined over Ω. Since Ω is partitioned into $\mathcal{F}$ and $\mathcal{L}$, the lower bound of $\mathcal{L}$ can be used as an approximate upper bound for $\mathcal{F}$ when the lower bound for $\mathcal{L}$ is tight. Thus, for any given C_{max} and Q_h, the lower-bound function $\tau\sqrt{\alpha} = \sqrt{C_{max}} - \sqrt{2Q_h}$ provides the network designer with a simple formula to estimate α

without seeking its close-form expression as a function of τ and C_{max}, which is impossible to obtain (due to the non-linearity of Eq. (2.17)). Furthermore, since the lower-bound function $\tau\sqrt{\alpha} = \sqrt{C_{max}} - \sqrt{2Q_h}$, which divides $\mathcal{F}$ and $\mathcal{L}$, is obtained by the constraint $Q_{max} \le C_{max}$. Letting $Q_{max} = C_{max}$, we get $Q_{max} = (\tau\sqrt{\alpha} + \sqrt{2Q_h})^2$, which can be used to estimate Q_{max} when the bound is tight. **(3)** Another interesting fact revealed by Theorem 2.4.2. is that Q_{max} is virtually independent of the multicast-tree bottleneck bandwidth μ since neither the lossless transmission condition/region nor the lower bound of $\mathcal{L}$ contains μ. This is not surprising since it is the relative difference between $R(t)$ and μ, instead of the absolute value of μ, that determines Q_{max}.

To illustrate the tightness of the derived lower bound of $\mathcal{L}$, the exact border which partitions Ω, the lower-bound function of $\mathcal{L}$ given by $K = \tau\sqrt{\alpha} = \sqrt{C_{max}} - \sqrt{2Q_h}$, and the configurations of the lossless transmission region $\mathcal{F}$ (the shaded area separated by $\tau\sqrt{\alpha} = \sqrt{C_{max}} - \sqrt{2Q_h}$) and lossy transmission region $\mathcal{L}$ are plotted in Figure 2.4, with $C_{max} = 400$ cells and Q_h=50 cells, which gives $K = 10$, and $\mu = 367$ cell/ms (about 155 Mbps). The exact border between $\mathcal{F}$ and $\mathcal{L}$ is obtained numerically (by solving Eq. (2.17) which needs μ). The lower-bound function of $\mathcal{L}$ (given by $K = \sqrt{C_{max}} - \sqrt{2Q_h} = \tau\sqrt{\alpha}$) plotted in Figure 2.4 is found to be very close to the exact border between $\mathcal{L}$ and $\mathcal{F}$. In addition, the smaller α, the tighter the bound is, which is consistent with the approximation $\log x \approx x - 1$ when x is close to 1 (see Eq. (A.7)).

2.4.2 The Second-Order Rate Control

As suggested by Theorem 2.4.2, α can be controlled to confine Q_{max} to C_{max}, and as long as $C_{max} > 2Q_h$, lossless transmission can be guaranteed by adjusting α in response to the variation of τ. The control over $\alpha = \frac{dR(t)}{dt}$ — which we call *α-control* — is the second-order control process which will be elaborated on below from a control-theoretic viewpoint. The original ATM recommendation for unicast (CI-based) ABR flow control is based on the Additive Increase and Multiplicative Decrease (AIMD) rate control algorithm. The

AIMD algorithm adapts the source rate $R(t)$ to the currently available bandwidth μ based on the feedback congestion information contained in CI-bit in feedback RM cell. Since the AIMD algorithm applies direct control over the rate $R(t)$ to match the target bandwidth μ, we can call AIMD the speed feedback-control process (from a control-theoretic viewpoint). The speed feedback-control system is traditionally called the first-order feedback control system which has one pole, or can be represented in a one-dimensional state-space. The α-control is an acceleration feedback-control process, which is one-order higher than the AIMD algorithm, since it exerts direct control over $\alpha = \frac{dR(t)}{dt}$. The acceleration feedback-control system is conventionally called the second-order feedback control system, which has two poles, or can be represented in a two-dimensional state-space. Thus, we also call the α-control the second-order rate control, which in fact provides one more dimension in state-space control over the dynamics of the proposed flow-control system.

2.4.3 The α-Control

The α-control is a discrete-time control process since it is only exercised when the source rate control is in a "decrease-to-increase" transition based on the buffer congestion feedback signal BCI. $BCI(n) := 0$ (or 1) if $Q_{max}^{(n)} \leq Q_{goal}$ (or $Q_{max}^{(n)} > Q_{goal}$), where Q_{goal} ($Q_h < Q_{goal} < C_{max}$) is the target buffer occupancy (also called a *setpoint*) in the equilibrium state. If the multicast-tree bottleneck shifts from a shorter path to a longer one, then τ will increase, making Q_{max} larger. When Q_{max} eventually grows beyond Q_{goal}, the buffer will tend to overflow, implying that the current α is too large for the increased τ. The source must reduce α to prevent cell losses. On the other hand, if τ decreases from its current value due to the shift of the multicast-tree bottleneck from a longer path to a shorter one, then Q_{max} will decrease. When $Q_{max} < Q_{goal}$, only a small portion of buffer space will be utilized, implying that the current α is too small for the decreased τ. The source should increase α to avoid buffer under-utilization and improve responsiveness in grabbing available bandwidth. So, feedback BCI contains the information on RM-cell RTT variations. Keeping $Q_h < Q_{goal} < C_{max}$ has two benefits: (1) the source can quickly grab available bandwidth; (2) it can achieve high throughput and network resource utilization.

The main purpose of α-control is to handle the buffer congestion resulting from the variation of τ. We set three goals for α-control: (1) ensure that $Q_{max}^{(n)}$ quickly converges to, and stays within, the neighborhood of Q_{goal}, which is upper-bounded by C_{max}, from an arbitrary initial value by driving their corresponding rate-gain parameters α_n to the neighborhood of α_{goal} for given τ; (2) maintain statistical fairness on the buffer occupancy among multiple multicast connections which share a common multicast-tree bottleneck; (3) minimize the extra cost incurred by the α-control algorithm. To achieve these goals, we propose a "converge-and-lock" α-control law in which the new value α_{n+1} is determined by α_n, and the feedback information BCI on Q_{max}'s current and one-step-old values, $Q_{max}^{(n)}$ and $Q_{max}^{(n-1)}$. The α-control law can be expressed by the following equations:

$$\alpha_{n+1} = \begin{cases} \alpha_n + p; & \text{if } BCI(n-1,n) = (0,0), \quad (Q_{max}^{(n-1)} \leq Q_{goal} \wedge Q_{max}^{(n)} \leq Q_{goal}) \\ q\alpha_n; & \text{if } BCI(n) = 1, \quad (Q_{max}^{(n)} > Q_{goal}) \\ \alpha_n/q; & \text{if } BCI(n-1,n) = (1,0), \quad (Q_{max}^{(n-1)} > Q_{goal} \wedge Q_{max}^{(n)} \leq Q_{goal}) \end{cases} \tag{2.5}$$

where q is the α-decrease factor such that $0 < q < 1$ and p is the α-increase step-size, whose values will be discussed next.

2.4.4 The Convergence Properties of the α-Control

To characterize the α-control's convergence properties, we first introduce the following two definitions.

Definition 2.4.2 *The neighborhood of target buffer occupancy, denoted by Q_{goal}, is specified by $\{Q_{goal}^l, Q_{goal}^h\}$ with*

$$Q_{goal}^l \triangleq \max_{n \in \{0,1,2,\cdots\}} \{Q_{max}^{(n)} \mid Q_{max}^{(n)} \leq Q_{goal}\} \tag{2.6}$$

$$Q_{goal}^h \triangleq \min_{n \in \{0,1,2,\cdots\}} \{Q_{max}^{(n)} \mid Q_{max}^{(n)} \geq Q_{goal}\} \tag{2.7}$$

where $Q_{max}^{(n)}$ is governed by the proposed α-control law. ■

Definition 2.4.3 $\{Q_{max}^{(n)}\} \triangleq \{Q_{max}(\alpha_n)\}$ *is said to monotonically converge to Q_{goal}'s neighborhood at time $n = n^*$ from its initial value $Q_{max}^{(0)} = Q_{max}(\alpha_0)$, if $BCI(0, 1, 2, 3, \cdots, n^*-1, n^*, n^*+1, n^*+2, n^*+3, \cdots) = (0, 0, 0, 0, \cdots, 0, 1, 0, 1, 0, \cdots)$ for $\alpha_0 < \alpha_{goal}$; and $BCI(0, 1, 2, 3, \cdots, n^*-1, n^*, n^*+1, n^*+2, n^*+3, \cdots) = (1, 1, 1, 1, \cdots, 1, 0, 1, 0, 1, \cdots)$ for $\alpha_0 > \alpha_{goal}$.* ■

The α-control is applied either in *transient* state, during which $Q_{max}^{(n)}$ has not yet reached Q_{goal}'s neighborhood, or in *equilibrium* state, in which $Q_{max}^{(n)}$ fluctuates within Q_{goal}'s neighborhood periodically. The α-control aims at making $Q_{max}^{(n)}$ converge rapidly in transient state and staying steadily within its neighborhood in equilibrium state. The following theorem summarizes the α-control law's convergence properties, optimal control conditions, and the method of computing the α-control parameters in both the transient and equilibrium states. Note that Q_{goal}^l and Q_{goal}^h are the closest attainable points around Q_{goal}, but Q_{goal} may not necessarily be the midpoint between Q_{goal}^l and Q_{goal}^h. The actual location of Q_{goal} between Q_{goal}^l and Q_{goal}^h depends on all rate-control parameters and the initial value of α_0.

Theorem 2.4.3 *Consider the proposed α-control law Eq. (2.5) which is applied to a multicast connection with its multicast-tree bottleneck characterized by Q_{goal}, Q_h, and τ. If (1) $\alpha = \alpha_0$, an arbitrary initial value at time $n = 0$, (2) $0 < q < 1$, and (3) $p \leq \left(\frac{1-q}{q}\right)\left(\frac{\sqrt{Q_{goal}} - \sqrt{2Q_h}}{\tau}\right)^2$, then the following claims hold:*

<u>**Claim 1:**</u> *During the transient state, the α-control law guarantees $Q_{max}^{(n)}$ to monotonically converge to Q_{goal}'s neighborhood $\{Q_{goal}^l, Q_{goal}^h\} = \{Q_{max}(\alpha_{goal}^l), Q_{max}(\alpha_{goal}^h)\}$,*

which are determined by

$$Q^l_{goal} = \begin{cases} Q_{max}\,(q^{n^*}\alpha_0); & \textit{if } \alpha_0 > \alpha_{goal} \\ Q_{max}\,(q(n^*p+\alpha_0)); & \textit{if } \alpha_0 \le \alpha_{goal} \end{cases} \tag{2.8}$$

$$Q^h_{goal} = \begin{cases} Q_{max}\,(q^{(n^*-1)}\alpha_0); & \textit{if } \alpha_0 > \alpha_{goal} \\ Q_{max}\,(n^*p+\alpha_0); & \textit{if } \alpha_0 \le \alpha_{goal} \end{cases} \tag{2.9}$$

where n^* is defined in Definition 2.4.3;

<u>**Claim 2:**</u> *During the* ***equilibrium state****, the fluctuation amplitudes of $Q^{(n)}_{max}$ around Q_{goal} are upper-bounded as follows:*

$$Q^h_{goal} - Q_{goal} \le \tau^2\alpha_{goal}\left(\frac{1}{q}-1\right) + \tau\sqrt{8\alpha_{goal}Q_h}\left(\frac{1}{\sqrt{q}}-1\right) \tag{2.10}$$

$$Q_{gaol} - Q^l_{goal} \le \tau^2\alpha_{goal}(1-q) + \tau\sqrt{8\alpha_{goal}Q_h}(1-\sqrt{q}) \tag{2.11}$$

and the diameter of neighborhood for the target buffer occupancy Q_{goal} is upper-bounded as follows:

$$Q^h_{gaol} - Q^l_{goal} \le \tau^2\alpha_{goal}\left(\frac{1}{q}-q\right) + \tau\sqrt{8\alpha_{goal}Q_h}\left(\frac{1}{\sqrt{q}}-\sqrt{q}\right) \tag{2.12}$$

where α_{goal} is the rate-gain parameter corresponding to Q_{goal} for given τ.

Proof. The proof is provided in Appendix C. ■

Remarks: The α-control law is similar to, but differs from, additive-increase/multiplicative-decrease algorithm in the following sense. In the transient state, the α-control law behaves like an additive-increase/multiplicative-decrease algorithm, which accommodates statistical convergence to fairness of buffer utilization among the multiple multicast connections sharing a common multicast-tree bottleneck. On the other hand, in equilibrium state, the α-control law guarantees buffer occupancy to be locked within its setpoint region at the first time when $Q^{(n)}_{max}$ reaches Q_{goal}'s neighborhood, regardless of the initial value α_0. In contrast, the additive-increase/multiplicative-decrease does not guarantee this monotonic convergence since α-control is a discrete-time control process and its convergence is dependent on α_0. The monotonic convergence ensures that $Q_{max}^{(n)}$ quickly converges to, and stays within, the neighborhood of its target value Q_{goal}. The extra cost paid for achieving these benefits is minimized since only a binary bit, BCI, is conveyed from the network bottleneck and two bits are used to store the current, and one-step-old feedback information, $BCI(n-1)$ and $BCI(n)$, at the source. The α-increase step-size p specified by condition (3) in Theorem 2.4.3 is a function of α-decrease factor q. A large q (small decrease step-size) requests a small p for the monotonic convergence. By the condition (3) of Theorem 2.4.3, if $q \to 1$, then $p \to 0$, which is expected since for a stable convergent system, zero decrease corresponds to zero increase in system state. According to Eqs. (2.10), (2.11), and (2.12), when $q \to 1$, both Q_{goal}^{l} and $Q_{goal}^{h} \to Q_{goal}$, i.e., $Q_{max}^{(n)}$'s fluctuation amplitude approaches zero, which also makes sense since $q \to 1$ implies $p \to 0$, thus $Q_{max}^{(n)}$ approaches a constant for all n.

To balance $R(t)$'s increase and decrease rates, and to ensure the average of the offered traffic load not to exceed the bottleneck bandwidth, each time when α_n is updated by the α-control law specified by Eq. (2.5), the proposed algorithm also updates the rate-decrease factor by $\beta_n = 1 - \frac{\alpha_n}{\mu}\Delta$ accordingly.

2.5 Single-Connection Bottleneck Dynamics

2.5.1 Equilibrium-State Analysis

The system is said to be in the equilibrium state if source rate $R(t)$ and multicast-tree bottleneck's $Q(t)$ have already converged to a certain regime and oscillate with a constant frequency and a steady average amplitude. The equilibrium-state analysis is mainly used to characterize the dynamics of the *multicast-tree bottleneck* after it has converged to a particular path and becomes relatively steady. In the equilibrium state, the source rate $R(t)$ fluctuates around the multicast-tree bottleneck's available bandwidth μ, and its $Q_{max}^{(n)}$ around Q_{goal}. The fluctuation amplitudes and periods are determined by the rate-control

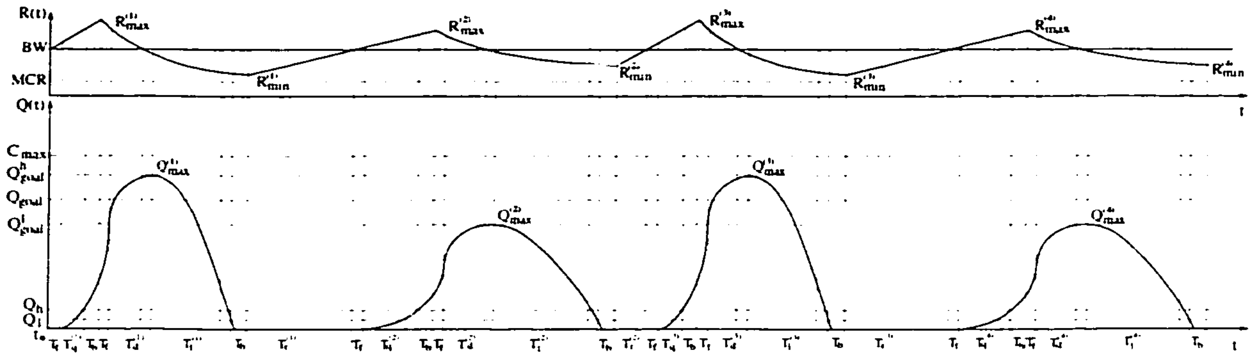


Figure 2.5: Dynamic behavior of $R(t)$ and $Q(t)$ for a single multicast connection.

parameters α, β; the multicast-tree bottleneck link's available bandwidth μ; its target buffer occupancy Q_{goal}; α-control parameters p, q; its congestion detection thresholds Q_h, Q_l, and delays T_b, T_f. To simplify the analysis of equilibrium state, we assume that the α-control parameters (i.e., α_0, Q_{goal}, p, and q) are properly selected according to the conditions specified in Theorem 2.4.3, such that $Q_{max}^{(n)}$ converges to a symmetric neighborhood of Q_{goal} where $Q_{goal} = \frac{1}{2}(Q_{goal}^l + Q_{goal}^h)$ and $Q_{goal}^h < C_{max}$.

Figure 2.5 illustrates the first 4 cycles of rate fluctuation and the associated queue-length function at the bottleneck link in equilibrium state with $\alpha_1 = \alpha_{goal}^h$. At time t_0, the rate reaches the link bandwidth μ (BW) and the queue starts to build up after a delay of T_f. At time $t_0 + T_b + T_q^{(1)}$, $Q(t)$ reaches Q_h and bandwidth congestion is detected. After a backward delay of T_b, the source receives $CI = 1$ feedback and its rate begins to decrease exponentially. $Q(t)$ reaches the peak as $R(t)$ drops back to the link bandwidth μ. When the rate falls below the link bandwidth, $Q(t)$ starts to decrease. After a time period of T_l elapsed, $Q(t)$ reaches Q_l, then the non-congestion condition ($CI = 0$) is detected and sent backward to the source. After a delay of T_b, the ($CI = 0$) feedback arrives at the source, then the "rate-decrease to rate-increase" transition condition ($local\_CI = 1 \wedge CI = 0$) is detected at the source. Subsequently, the source adjusts the next rate-gain parameter α_2 to a smaller value, $q\alpha_1$ (β_2 is also adjusted accordingly by $\beta_2 = 1 - \frac{\alpha_2}{\mu}\Delta$) since $BCI(1) = 1$ (due to $Q_{max}^{(1)} > Q_{goal}$) is received in the feedback RM cell. Then, the source rate increases linearly with the newly updated rate-gain parameter $\alpha_2 = q\alpha_1 = \alpha^l_{goal}$. When $R(t)$ reaches μ after a time period of $T_r^{(1)}$, the system starts the second fluctuation cycle.

The dynamic behavior of the second cycle of fluctuation follows a similar pattern to that in the first cycle except for the adjusted rate-control parameters α_2 and β_2 resulting in a longer cycle length due to smaller increase/decrease rates. When the transition from rate-decrease to rate-increase is detected again for the second fluctuation cycle, the source sets $\alpha_3 = \alpha_2/q$ because $Q^{(2)}_{max} < Q_{goal}$, i.e., $BCI(2) = 0$, hence $BCI(1,2) = (1,0)$. But $\alpha_3 = \alpha_2/q = (q\alpha_1)/q = \alpha_1$ since α_n has already converged to $\{\alpha^l_{goal}, \alpha^h_{goal}\}$ in equilibrium state. Thus, the dynamic behavior of the third fluctuation cycle is exactly the same as the first cycle. Likewise, the fourth cycle is the same as the second one, and so on. So, we can only focus on the dynamic behavior of the first fluctuation cycle $T_1 = 2(T_f + T_b) + T_q^{(1)} + T_d^{(1)} + T_l^{(1)} + T_r^{(1)}$ and the second fluctuation cycle $T_2 = 2(T_f + T_b) + T_q^{(2)} + T_d^{(2)} + T_l^{(2)} + T_r^{(2)}$. We define the *control period* to be $T = T_1 + T_2$.

In the i-th fluctuation cycle ($i = 1, 2$), let $R^{(i)}_{max}$ and $R^{(i)}_{min}$ be its maximum and minimum rates, respectively, and $Q^{(i)}_{max}$ be its maximum queue length, then we have

$$R^{(i)}_{max} = \mu + \alpha_i \left(T_q^{(i)} + T_b + T_f \right) \tag{2.13}$$

where $T_q^{(i)} = \sqrt{\frac{2Q_h}{\alpha_i}}$ is the time for the queue length to grow from 0 to Q_h, $\alpha_1 = \alpha^h_{goal} = \alpha^l_{goal}/q$ and $\alpha_2 = q\alpha_1 = \alpha^l_{goal}$. For convenience of presentation, we define

$$T^{(i)}_{max} \triangleq T_b + T_q^{(i)} + T_f = T_b + \sqrt{\frac{2Q_h}{\alpha_i}} + T_f \tag{2.14}$$

which is the time for $R(t)$ to increase from μ to its maximum $R^{(i)}_{max}$ by exercising linear rate-increase control. Then, the maximum queue length is expressed as

$$Q^{(i)}_{max} = \int_0^{T^{(i)}_{max}} \alpha_i t \; dt + \int_0^{T_d^{(i)}} \left(R^{(i)}_{max} e^{-(1-\beta_i)\frac{t}{\Delta}} - \mu \right) \, dt \tag{2.15}$$

where $T_d^{(i)}$ is the time for $R(t)$ to drop from $R^{(i)}_{max}$ back to μ (i.e., the BW, see Figure 2.5), and is obtained, by letting $R(T_d^{(i)}) = \mu$, as

$$T_d^{(i)} = -\frac{\Delta}{(1-\beta_i)} \log \frac{\mu}{R^{(i)}_{max}} \tag{2.16}$$

Thus, we obtain

$$Q_{max}^{(i)} = \frac{\alpha_i}{2}\left[T_{max}^{(i)}\right]^2 + \frac{\Delta}{1-\beta_i}\left[\alpha_i T_{max}^{(i)} + \mu \log \frac{\mu}{R_{max}^{(i)}}\right]. \tag{2.17}$$

Letting $T_l^{(i)}$ be the period for $Q(t)$ to decrease from $Q_{max}^{(i)}$ to Q_l, we have

$$Q_{max}^{(i)} - Q_l = \int_0^{T_l^{(i)}} \mu\left(1 - e^{-(1-\beta_i)\frac{t}{\Delta}}\right) dt. \tag{2.18}$$

So, $T_l^{(i)}$ is the non-negative real root of non-linear equation:

$$e^{-(1-\beta_i)\frac{T_l^{(i)}}{\Delta}} + \frac{1-\beta_i}{\Delta}\left[T_l^{(i)} - \frac{Q_{max}^{(i)} - Q_l}{\mu}\right] - 1 = 0. \tag{2.19}$$

Then, the minimum rate is given by $R_{min}^{(i)} = \mu e^{-(1-\beta_i)\frac{T_l^{(i)} + T_b + T_f}{\Delta}}$.

The control period is determined by

$$T = \sum_{i=1}^{2} T_i = \sum_{i=1}^{2}\left[T_q^{(i)} + T_d^{(i)} + T_l^{(i)} + 2\tau + T_r^{(i)}\right] \tag{2.20}$$

where $T_r^{(i)} = \left(\mu - R_{min}^{(i)}\right)/\alpha_{i+1}$ is the time for $R(t)$ to grow from $R_{min}^{(i)}$ to μ with the increase-rate parameter α_{i+1} ($\alpha_3 = \alpha_1$). Note that each T_i contains two RTTs, which correspond to the two transitions of $R(t)$ (from linear to exponential and then back to linear).

The average equilibrium throughput, denoted by $\overline{R}$, can be calculated by averaging $R(t)$ over control period T as follows

$$\overline{R} = \frac{1}{T}\sum_{i=1}^{2}\left[\int_0^{T_{max}^{(i)}} (\mu + \alpha_i t)\, dt + \int_0^{T_e^{(i)}} \left(R_{max}^{(i)} e^{-(1-\beta_i)\frac{t}{\Delta}}\right) dt + \int_0^{T_r^{(i)}} \left(R_{min}^{(i)} + \alpha_{i+1} t\right) dt\right] \tag{2.21}$$

where $T_e^{(i)} = T_d^{(i)} + T_l^{(i)} + \tau$ is the time spent on exponential-decrease rate control within the i-th cycle. The above equation is reduced to:

$$\overline{R} = \frac{1}{T}\sum_{i=1}^{2}\left[\mu T_{max}^{(i)} + \frac{\alpha_i}{2}[T_{max}^{(i)}]^2 + R_{max}^{(i)}\left(\frac{\Delta}{1-\beta_i}\right)\left(1 - e^{-(1-\beta_i)\frac{T_e^{(i)}}{\Delta}}\right) + T_r^{(i)} R_{min}^{(i)} + \frac{\alpha_{i+1}}{2}[T_r^{(i)}]^2\right] \tag{2.22}$$

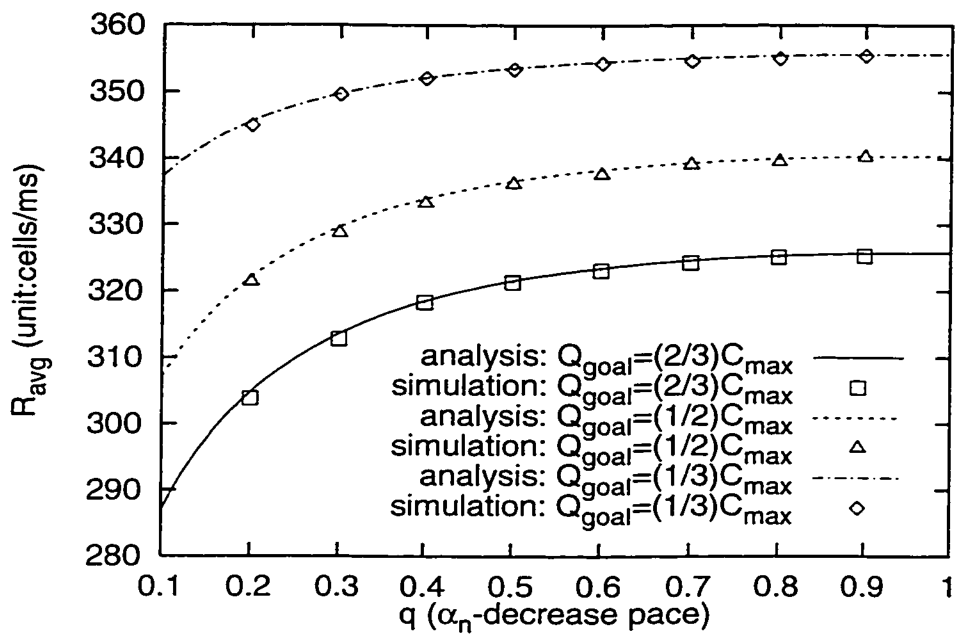


Figure 2.6: Equilibrium-state performance evaluation: average throughput $\overline{R}$ vs. q

2.5.2 Equilibrium-State Performance Evaluation

Assume (i) the bottleneck link bandwidth μ = 155 Mbps (367 cells/ms) and C_{max} = 750 cells, and (ii) the bottleneck is detected at a node farthest away from the source, so, $T_b = T_f = 1$ ms and $\tau = T_b + T_f = 2$ ms. Also, we use $\Delta = 0.5\tau = 1$ ms, $Q_h = 50$ cells, $Q_l = 25$ cells, and the initial source rate $R_0 = \mu$ as we are dealing with equilibrium state.

Figure 2.6 plots $\overline{R}$ vs. q for different values of Q_{goal}, which are obtained from the analysis and the simulations.[8] We first focus on the ideal case where $Q_{goal} = \frac{1}{2}(Q^h_{goal} + Q^l_{goal})$, i.e., $Q^{(n)}_{max}$ fluctuates symmetrically above and below Q_{goal}. Figure 2.6 shows that $\overline{R}$ monotonically increases as q grows from 0.1 to 1.0. This is expected since a smaller q leads to a larger fluctuation of $R^{(n)}_{max}$ and $Q^{(n)}_{max}$, which defeats the equilibrium-state performance of $\overline{R}$. When q gets larger, the fluctuation amplitudes of $Q^{(n)}_{max}$ and $R^{(n)}_{max}$ get smaller, as shown in Theorem 2.4.3. In the extreme case when $q \to 1$ (q cannot be equal to 1 since $q = 1$ means that the α-control is shut down), $R^{(n)}_{max}$ approaches a constant value, and the

[8]The simulations were performed by using the NetSim package [31], and for comparison purposes, the parameters were set exactly the same as those used the analysis.

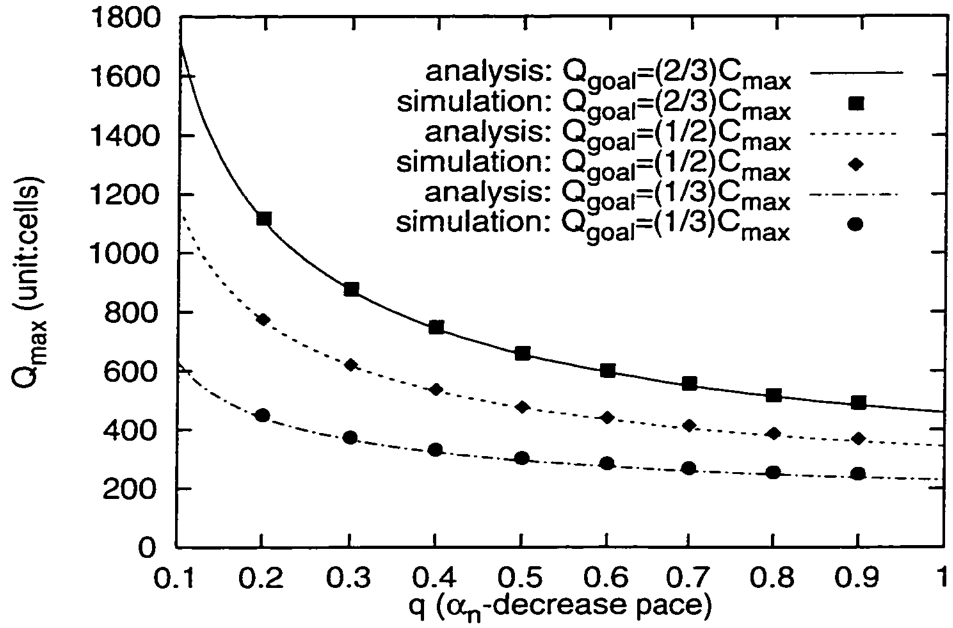


Figure 2.7: Equilibrium-state performance evaluation: maximum queue length Q_{max} vs. q

equilibrium-state performance of $\overline{R}$ attains its maximum. Figure 2.6 also indicates that for the same value of q, a smaller value of $Q_{goal} = kC_{max}$, $0 < k < 1$, leads to a larger $\overline{R}$ in equilibrium state, which is also consistent with our observations in [7], since a smaller Q_{goal} implies a smaller α_{goal}. In summary, Figure 2.6 shows (i) an increasingly sharp drop in $\overline{R}$ when q gets smaller than 0.4, and (ii) a slow gain in $\overline{R}$ when $q > 0.6$, providing information on how to select q for the α-control to operate in a balanced region within which an optimal balance between average throughput and response speed is achieved. In addition, Figure 2.6 shows that the analytical results based on the fluid modeling fit the simulation results well. The slight discrepancy is due to the RM-cell processing and queuing delays, and the fluid analysis approximation.

Although Q_{goal} can be anywhere between Q^l_{goal} and Q^h_{goal}, depending on α_0, in order to analyze how q affects the maximum buffer requirement, we consider the worst case when $Q_{goal} \gtrsim Q^l_{goal}$. Figure 2.7 plots Q_{max} vs. q in the worst case of buffer requirement. Q_{max} is observed to increase as q decreases, which makes sense since a smaller q implies a larger fluctuation amplitude of $Q_{max}^{(n)}$. Moreover, when q is very small, particularly below the range of 0.4–0.6, Q_{max} shoots up quickly. Also, when q is beyond the range of 0.4–0.6, Q_{max} drops slowly as q increases. Again, we observe that the analytical results are verified by the simulation results, since the latter closely matches the former in terms of the maximum buffer requirement, as shown in Figure 2.7

2.5.3 Transient-State Analysis

An equilibrium state can be broken by either the change of the multicast-tree bottleneck from one path to another with different flow control parameters; or the change of available bandwidth due to the variation of cross traffic or the number of active VCs (Virtual Circuits). After an equilibrium state is broken, the system experiences a certain period of transient state, during which the system typically converges to a new equilibrium state if any. Thus, the transient-state analysis is mainly targeted to characterize the system dynamics while the multicast-tree bottleneck path is still in progress of changing or converging to a path; or the bottleneck's available bandwidth for this multicast connection is changing due to the variation of cross-traffic. Here, the transient-state analysis mainly focuses on the case where the system's entry to the transient state is caused by the change of the multicast-tree bottleneck from one path to another with the different RTTs (τ). The system can move to transient state due to the variation of RTT τ in two different cases: (I) $\alpha_0 > \alpha_{goal}^h$, the rate convergence is underdamped, and (II) $\alpha_0 < \alpha_{goal}^l$, the rate convergence is overdamped, where α_{goal}^h and α_{goal}^l are functions of Q_{goal}, p, q, τ, and μ.

Denote the rate-gain parameter at the beginning of transient state by α_0. Let the new multicast-tree bottleneck's target rate-gain parameter be $\widetilde{\alpha_{goal}}$ which corresponds to the new multicast-tree bottleneck path's RM-cell RTT $\widetilde{\tau}$ and target bandwidth $\widetilde{\mu}$. The following theorem gives a formula to calculate the number of transient cycles.

Theorem 2.5.1 *Consider a multicast-tree bottleneck characterized by Q_{goal}, Q_h, p, and q. If the initial rate-gain parameter $\alpha = \alpha_0$, the new RM-cell RTT $\tau = \widetilde{\tau}$, and new target* *bandwidth $\mu = \widetilde{\mu}$, then the number of transient cycles, N, is determined by*

$$N = \begin{cases} \left\lfloor \log\left(\frac{\widetilde{\alpha_{goal}}}{\alpha_0}\right) / \log q \right\rfloor; & \text{if } \alpha_0 > \widetilde{\alpha_{goal}} \\ \left\lceil (\widetilde{\alpha_{goal}} - \alpha_0)/p \right\rceil; & \text{if } \alpha_0 \le \widetilde{\alpha_{goal}} \end{cases} \tag{2.23}$$

where $\widetilde{\alpha_{goal}}$ is the non-negative real root of non-linear equation:

$$\frac{\widetilde{\alpha_{goal}}}{2}\left(\widetilde{\tau} + \sqrt{\frac{2Q_h}{\widetilde{\alpha_{goal}}}}\right)^2 + \widetilde{\mu}\left(\widetilde{\tau} + \sqrt{\frac{2Q_h}{\widetilde{\alpha_{goal}}}}\right) + \frac{\widetilde{\mu}^2}{\widetilde{\alpha_{goal}}} \log \frac{\widetilde{\mu}}{\widetilde{\mu} + \widetilde{\alpha_{goal}}\left(\widetilde{\tau} + \sqrt{\frac{2Q_h}{\widetilde{\alpha_{goal}}}}\right)} - Q_{goal} = 0. \tag{2.24}$$

and can be approximated as

$$\widetilde{\alpha_{goal}} \approx \left(\frac{\sqrt{Q_{goal}} - \sqrt{2Q_h}}{\widetilde{\tau}}\right)^2, \tag{2.25}$$

if Q_{goal} is small.

Proof. The proof is provided in Appendix E. ■

Let $R_{peak}^{(i)}$ and $Q_{peak}^{(i)}$ be the peak source rate and queue length, respectively, in the i-th transient cycle, $i = 1, 2, \cdots, N(\ge 1)$ (by assuming $\alpha_0 \ge \frac{1}{q}\widetilde{\alpha_{goal}}$ or $\alpha_0 \le \widetilde{\alpha_{goal}} - p$). Let's start from the first transient cycle, or $i = 1$. Since the rate-increase function in the first transient cycle is $R(t) = R_0 + \alpha_0 t$, we have

$$R_{peak}^{(1)} = R_0 + \alpha_0(T_q^{(1)} + \widetilde{\tau}) \tag{2.26}$$

where $T_q^{(1)} = \frac{1}{\alpha_0}\left[-(R_0 - \widetilde{\mu}) + \sqrt{(R_0 - \widetilde{\mu})^2 + 2\alpha_0 Q_h}\right]$ is obtained by solving the following equation:

$$Q_h = \int_0^{T_q^{(1)}} (R(t) - \widetilde{\mu})\, dt. \tag{2.27}$$

For convenience, let $T_{peak}^{(1)} \triangleq T_q^{(1)} + \widetilde{\tau}$ be the time for $R(t)$ to increase from R_0 to $R_{peak}^{(1)}$. Then, the peak queue length can be obtained as:

$$Q_{peak}^{(1)} = \int_0^{T_{peak}^{(1)}} (R_0 + \alpha_0 t - \widetilde{\mu})\, dt + \int_0^{T_d^{(1)}} \left(R_{peak}^{(1)} e^{-(1-\beta_0)\frac{t}{\Delta}} - \widetilde{\mu}\right) dt \tag{2.28}$$

where $T_d^{(1)} = -\frac{\Delta}{(1-\beta_0)} \log \frac{\widetilde{\mu}}{R_{peak}^{(1)}}$ is the time for $R(t)$ to drop from R_{peak} back to $\widetilde{\mu}$. Reducing Eq. (2.28) gives

$$Q_{peak}^{(1)} = (R_0 - \widetilde{\mu})T_{peak}^{(1)} + \frac{\alpha_0}{2}\left[T_{peak}^{(1)}\right]^2 + \frac{\Delta}{1-\beta_0}\left[\alpha_0 T_{peak}^{(1)} + (R_0 - \widetilde{\mu}) + \widetilde{\mu} \log \frac{\widetilde{\mu}}{R_{peak}^{(1)}}\right] \tag{2.29}$$

When $R_0 = \widetilde{\mu}$, Eq. (2.29) reduces to Eq. (2.17), which is consistent with the fact that $Q_{max}^{(i)}$ is the special case of $Q_{peak}^{(1)}$ with $R_0 = \widetilde{\mu}$.

To compute the first transient-state cycle, we need to find $T_l^{(1)}$ which is the non-negative real root of nonlinear equation:

$$e^{-(1-\beta_0)\frac{T_l^{(1)}}{\Delta}} + \left(\frac{1-\beta_0}{\Delta}\right)T_l^{(1)} - \left[\left(\frac{Q_{peak}^{(1)} - Q_l}{\widetilde{\mu}}\right)\left(\frac{1-\beta_0}{\Delta}\right) + 1\right] = 0. \tag{2.30}$$

The period of this transient-state cycle is

$$T^{(1)} = T_q^{(1)} + T_d^{(1)} + T_l^{(1)} + 2\widetilde{\tau} + T_r^{(1)} \tag{2.31}$$

where

$$T_r^{(1)} = \frac{\widetilde{\mu}}{\alpha_1}\left(1 - e^{-(1-\beta_0)\frac{T_l^{(1)}+\widetilde{\tau}}{\Delta}}\right) \tag{2.32}$$

is the time for $R(t)$ to reach $\widetilde{\mu}$ from its lowest value in the first transient cycle.

The average throughput during the first transient-state cycle is expressed by

$$\begin{aligned}\overline{R}^{(1)} \quad = \quad & \frac{1}{T^{(1)}}\Bigg[\; R_0 T_{peak}^{(1)} + \frac{\alpha_0}{2}[T_{peak}^{(1)}]^2 + R_{peak}^{(1)}\left(\frac{\Delta}{1-\beta_0}\right)\left(1 - e^{-(1-\beta_0)\frac{T_d^{(1)}+T_l^{(1)}+\widetilde{\tau}}{\Delta}}\right) \\ & + T_r^{(1)}\left(\widetilde{\mu}e^{-(1-\beta_0)\frac{T_l^{(1)}+\widetilde{\tau}}{\Delta}}\right) + \frac{\alpha_1}{2}[T_r^{(1)}]^2\Bigg]\end{aligned} \tag{2.33}$$

Now, let's consider cases for $2 \le i \le N$ (N is given by Eq. (2.23) of Theorem 2.5.1). Since the performance parameters are derived similarly to the case of $i = 1$, we only give the final expressions for the average throughput, the peak queue length, and the length of the i-th transient cycle ($2 \le i \le N$):

$$\overline{R}^{(i)} = \frac{1}{T^{(i)}}\Bigg[\tilde{\mu}T^{(i)}_{peak} + \frac{\alpha_{i-1}}{2}[T^{(i)}_{peak}]^2 + R^{(i)}_{peak}\left(\frac{\Delta}{1-\beta_{i-1}}\right)\left(1 - e^{-(1-\beta_{i-1})\frac{T_d^{(i)}+T_l^{(i)}+\tilde{\tau}}{\Delta}}\right)$$
$$+ T_r^{(i)}\left(\tilde{\mu}e^{-(1-\beta_{i-1})\frac{T_l^{(i)}+\tilde{\tau}}{\Delta}}\right) + \frac{\alpha_i}{2}[T_r^{(i)}]^2\Bigg] \tag{2.34}$$

$$Q^{(i)}_{peak} = \frac{\alpha_{i-1}}{2}[T^{(i)}_{peak}]^2 + \alpha_{i-1}\frac{\Delta}{(1-\beta_{i-1})}T^{(i)}_{peak} + \tilde{\mu}\frac{\Delta}{(1-\beta_{i-1})}\log\frac{\tilde{\mu}}{R^{(i)}_{peak}} \tag{2.35}$$

$$T^{(i)} = \sqrt{\frac{2Q_h}{\alpha_{i-1}}} + T_d^{(i)} + T_l^{(i)} + T_r^{(i)} + 2\tilde{\tau} \tag{2.36}$$

where

$$T^{(i)}_{peak} = \tilde{\tau} + \sqrt{\frac{2Q_h}{\alpha_{i-1}}}, \tag{2.37}$$

$$R^{(i)}_{peak} = \tilde{\mu} + \alpha_{i-1}T^{(i)}_{peak}, \tag{2.38}$$

$$T_d^{(i)} = \frac{-\Delta}{1-\beta_{i-1}}\log\frac{\tilde{\mu}}{R^{(i)}_{peak}}, \tag{2.39}$$

$$T_r^{(i)} = \frac{\tilde{\mu}}{\alpha_i}\left[1 - e^{-(1-\beta_{i-1})\frac{T_l^{(i)}+\tilde{\tau}}{\Delta}}\right] \tag{2.40}$$

and $T_l^{(i)}$ is the non-negative real root of the following non-linear equation:

$$e^{-(1-\beta_{i-1})\frac{T_l^{(i)}}{\Delta}} + \left(\frac{1-\beta_{i-1}}{\Delta}\right)T_l^{(i)} - \left[\left(\frac{Q^{(i)}_{peak} - Q_l}{\tilde{\mu}}\right)\left(\frac{1-\beta_{i-1}}{\Delta}\right) + 1\right] = 0. \tag{2.41}$$

The entire transient-state period is then $T_{tran} = \sum_{i=1}^{N} T^{(i)}$, and its average throughput is expressed by

$$\overline{R}_{tran} = \frac{1}{T_{tran}}\sum_{i=1}^{N}\overline{R}^{(i)}T^{(i)}. \tag{2.42}$$

The peak queue length for the case of $\alpha_0 > \alpha^h_{goal}$ is $Q_{peak} = Q^{(1)}_{peak}$. Here N is given by Eq. (2.23) of Theorem 2.5.1, and α_i is determined by the α-control law defined in Eq. (2.5).

2.5.4 Transient-State Performance Evaluation

Using the analytical results, we derived numerical solutions to evaluate transient-state performance. Assume the same flow-control parameter settings as in the equilibrium-state

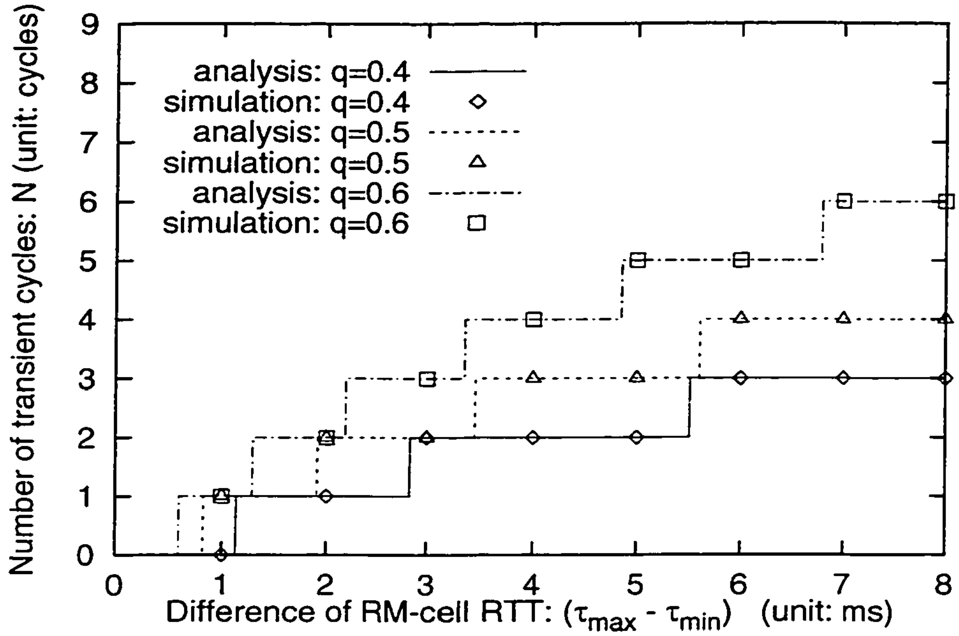


Figure 2.8: Transient-state performance evaluation: No. of Tran-cycles N vs. $(\tau_{max}-\tau_{min})$.

analysis, except that $C_{max} = 700$ cells and $Q_{goal} = \frac{1}{2}C_{max} = 350$ cells, and α_0 is specified by $\mu_0 = 367$ cells/ms and $\tau_0 = 2$ ms. To study the worst case, we let the initial $\tau_0 = \tau_{min} \triangleq \min_{i\in\{1,2,\cdots,n\}}\{\tau_i\}$ and $\widetilde{\tau} = \tau_{max} \triangleq \max_{i\in\{1,2,\cdots,n\}}\{\tau_i\}$ of a multicast VC with n paths. Also, assume $\widetilde{\mu} = 267$ cells/ms. Figure 2.8 plots N, obtained numerically by Eq. (2.23) and simulations using the NetSim [31], vs. $(\tau_{max} - \tau_{min})$ for different values of q. N is found to increase stepwise monotonically with $(\tau_{max} - \tau_{min})$. This is expected since a large variation in RM-cell RTT requires more transient cycles to converge to the new optimal equilibrium state. A smaller q results in a fewer number of transient cycles. Thus, q measures the speed of convergence. These observations have been exactly duplicated by simulations, thus verifying Theorem 2.5.1. Figure 2.9 shows the numerical and simulation results for Q_{peak} vs. $(\tau_{max} - \tau_{min})$ with Q_{goal} varying, where we assume $R_0 = 367$ cells/ms, $\widetilde{\mu} = 347$ cells/ms, $\tau_{min} = \tau_0 = 2$ ms, and $C_{max} = 700$ cells. Q_{peak} is observed to shoot up quickly with $(\tau_{max} - \tau_{min})$, further justifying the necessity of α-control, and a larger target buffer occupancy is found to result in a faster increase of Q_{peak}. The simulation results closely

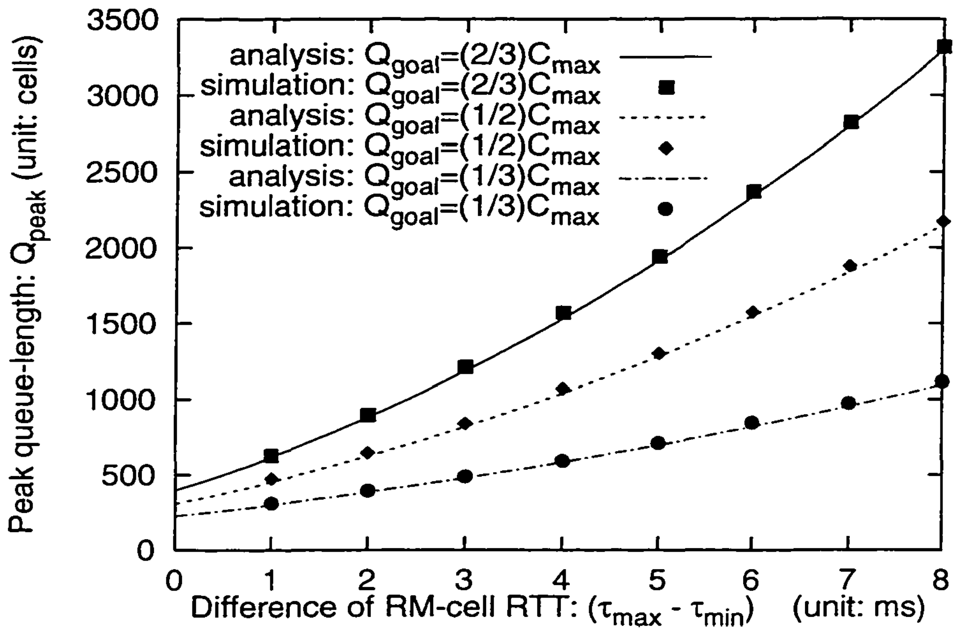


Figure 2.9: Transient-state performance evaluation: Peak que-length Q_{peak} vs. $(\tau_{max}-\tau_{min})$.

match the analytical results as shown in Figure 2.9.

2.5.5 The Greatest Lower Bound for the Target Buffer Occupancy

How to choose the target buffer occupancy Q_{goal} is a practically important design problem associated with the α-control. Usually, as long as Q_{goal} can ensure the full bandwidth utilization, a small Q_{goal} is desired, because a large Q_{goal} may increase queuing delay and delay variations, affecting the network dynamics and stability. Using the analytical results derived in Section 2.5.1, the theorem given below finds the greatest lower bound for Q_{goal} and its relationships with α, τ, and Q_h.

Theorem 2.5.2 *Consider a connection flow-controlled by the proposed rate-control scheme described by Eqs. (2.2) and (2.3). If (i) the upper queue-length threshold $Q_h < \frac{1}{2}\xi < \infty$, (ii) its RTT $\tau > 0$, and (iii) the rate-gain parameter α is controlled by the α-control law defined in Eq. (2.5), then the following claims hold:*

<u>**Claim 1:**</u> *The greatest lower bound of $Q_{goa}(\alpha_n, \tau)$ under the α-control defined in Eq. (2.5)*

exists and is determined by:

$$\inf_{\tau>0,\,\alpha_n>0,\,n=1,2,\cdots,\infty} \{Q_{goal}(\alpha_n,\tau)\} = 2Q_h; \tag{2.43}$$

<u>Claim 2:</u> *The right-hand limit of $Q_{goa}(\alpha,\tau)$ at $\alpha = 0$ in the continuous-domain of α exists and is determined by:*

$$\lim_{\alpha\downarrow 0} Q_{goal}(\alpha,\tau) = 2Q_h; \tag{2.44}$$

where all variables are the same as defined in Section 2.4.3 and Section 2.5.1.

Proof. The detailed proof is provided in Appendix F ∎

Remarks on Theorem 2.5.2: <u>Claim 1</u> derives the greatest lower bound of $Q_{goal}(\alpha,\tau)$ under the proposed α-control law, showing that Q_{goal} must be at least larger than $2Q_h$ for $\alpha > 0$ and $\tau > 0$. <u>Claim 2</u> shows that α must approach 0 for $Q_{goal}(\alpha,\tau)$ to converge to its greatest lower bound $2Q_h$. Combining <u>Claim 1</u> and <u>Claim 2</u>, we choose $Q_{goal} \gg 2Q_h$ as specified in Section 2.2. Theorem 2.5.2 also provides the network designer with an explicit guidance in selecting Q_h for any desired target buffer occupancy Q_{goal} and the given buffer capacity C_{max} at routers. As shown in [32], $Q_{max}(\alpha,\tau)$ increases as Q_h increases, and so does $Q_{goal}(\alpha,\tau)$. On the other hand, too small a Q_h is also undesirable because too small a Q_h may decrease the bandwidth utilization.

2.5.6 Packet-Loss Analysis

Since the buffer size at routers is always finite, in this section we focus on the case where packets are lost due to buffer overflow.

2.5.6.1 Packet-Loss Calculation

To quantitatively evaluate the loss-control performance of the proposed scheme, we introduce the following definition:

Definition 2.5.1 *The* ***packet-loss rate****, denoted by γ, is the percentage of the lost packets among all the transmitted packets and the* ***link-transmission efficiency****, denoted by η, is the fraction of packets successfully transmitted (without retransmitting them) among all packets transmitted. Then γ and η in one rate-control cycle are expressed as:*

$$\gamma \triangleq \frac{\rho}{T\overline{R}} \quad \textit{and} \quad \eta \triangleq 1-\gamma = 1 - \frac{\rho}{T\overline{R}} \tag{2.45}$$

where T is the rate-control cycle specified by Eq. (2.20), ρ is the number of lost packets during T, and $\overline{R}$ is the average throughput determined by Eq. (2.22). ■

The link-transmission efficiency η is an important metric for flow and error control since it measures the percentage of link bandwidth used by successfully-transmitted packets. The following theorem gives an explicit formula to calculate the number ρ of packet losses from which both η and γ can be derived.

Theorem 2.5.3 *If a connection with buffer capacity $Q_h < \xi < \infty$ is under the rate-control scheme described by the state equations (2.2)–(2.3) and the α-control law defined in Eq. (2.5), then the number, ρ, of lost packets during one rate-control cycle T is determined by:*

$$\rho = \begin{cases} \frac{1}{2}\,\alpha\left(T_{max}^2 - t_\xi^2\right) - \mu T_d + R_{max}\dfrac{\Delta}{1-\beta}\left[1 - e^{-\frac{1-\beta}{\Delta}T_d}\right]; & \textit{if } t_\xi \le T_{max} \\ \mu\left(t_\xi - T_{max} - T_d\right) + R_{max}\frac{\Delta}{1-\beta}\left[e^{-\frac{1-\beta}{\Delta}(t_\xi - T_{max})} - e^{-\frac{1-\beta}{\Delta}T_d}\right]; & \textit{if } t_\xi > T_{max} \end{cases} \tag{2.46}$$

where all variables are the same as defined in Section 2.5.1, except that $t_\xi = \sqrt{\frac{2\xi}{\alpha}}$ if $\xi \le \frac{1}{2}\alpha T_{max}^2$ (i.e., $t_\xi = \sqrt{\frac{2\xi}{\alpha}} \le T_{max}$, which determines the value of variable t_ξ for the condition used in the first part of Eq. (2.46)); else t_ξ is the non-negative real root of the following non-linear equation, which determines the value of variable t_ξ for the condition used in the

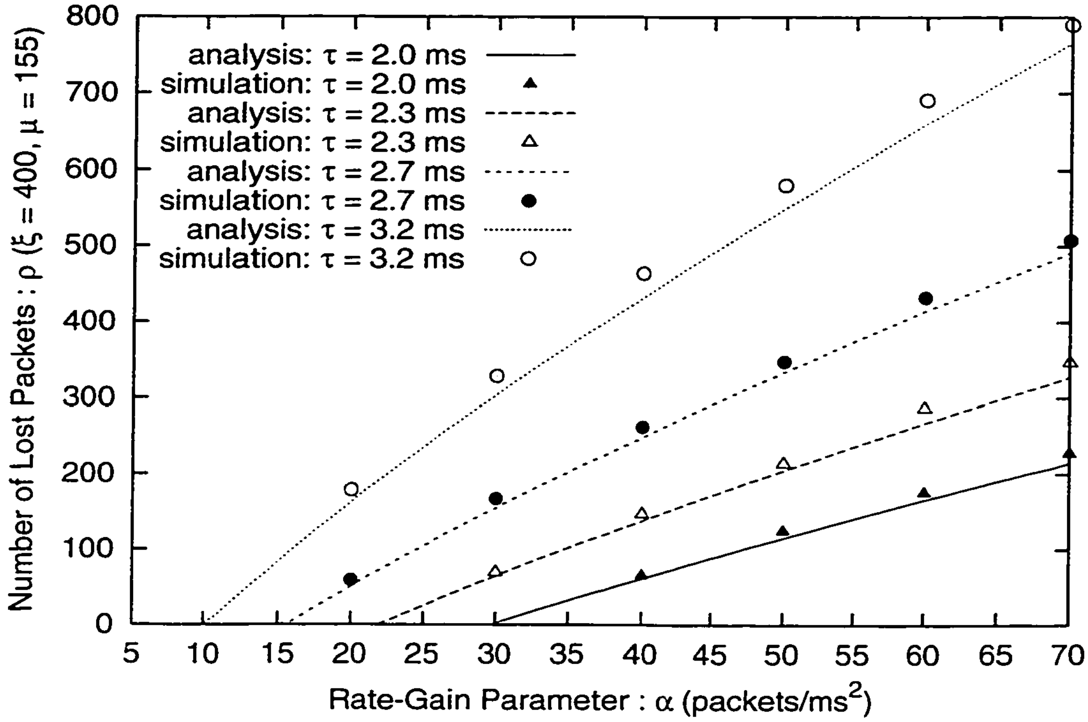


Figure 2.10: Number of lost packets (ρ) vs. α.

second part of Eq. (2.46):

$$\frac{1}{2}\alpha T_{max}^2 + R_{max}\frac{\Delta}{1-\beta}\left(1 - e^{-(1-\beta)\frac{t_\xi - T_{max}}{\Delta}}\right)$$
$$- \mu(t_\xi - T_{max}) - \xi = 0, \quad if\, \xi > \frac{1}{2}\alpha T_{max}^2. \tag{2.47}$$

Proof. The detailed proof is provided in Appendix H. ■

2.5.6.2 Performance Evaluation of Loss Control

Consider the bottleneck with μ = 367 packets/ms (155 Mbps), ξ = 400 packets; Q_h = 50 packets, and q = 0.6. Fig. 2.10 plots the number of lost packets, ρ, obtained from Eq. (2.46), against α for different RTTs τ's. Note that ρ increases with α, and for a given α, ρ gets larger as τ increases. It is therefore necessary to apply α-control to reduce the packet

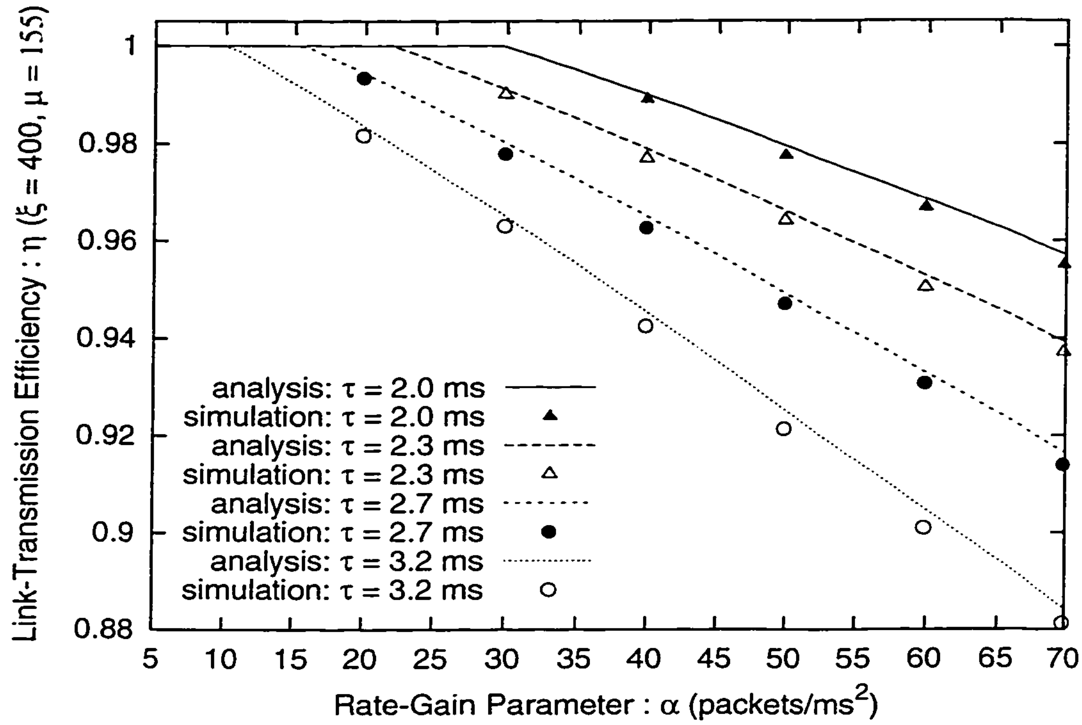


Figure 2.11: Link-transmission efficiency (η) vs. α.

losses due to the increase in the number and RTT of cross-traffic flows. Packet losses cause retransmissions, and thus affect link-transmission efficiency η. In Fig. 2.11, η is plotted against α for the same parameters. As illustrated in Fig. 2.11, $\eta = 1$ at the beginning, implying that there is no retransmission (loss) if α is controlled to be small enough under the α-control for any given τ. As α increases, Fig. 2.11 shows that η is a decreasing function of α, and drops faster for larger τ's. For instance, $\gamma = 1 - \eta \leq 2\%$ of packets need to be retransmitted if α is controlled to be smaller than 50 packets/ms^2 for $\tau = 2$ ms, but to keep $\eta \geq 98\%$ for $\tau = 3.2$ ms, α needs to be limited to no larger than 22 packets/ms^2. Using the NetSim [31], we also simulated packet losses and link-transmission efficiency, which agree well with the numerical results (see Figs. 2.10–2.11).

2.6 Multiple Multicast Connections

We now model and analyze the convergence properties of the proposed scheme for M (> 1) concurrent multicast connections that share a common multicast-tree bottleneck.

2.6.1 Efficiency and Fairness of the α-Control

Since $Q_{max}(\alpha)$ is a one-to-one mapping function between Q_{max} and α as shown in Eq. (2.17), buffer-allocation control can be handled equivalently by α-allocation control. We introduce the following criteria to evaluate the α-control law for buffer management in terms of α-allocation.

Definition 2.6.1 *Let vector $\alpha(k) = (\alpha_1(k), \cdots, \alpha_n(k))$ be the rate-gain parameters at time k for n multicast connections sharing a common bottleneck characterized by $\alpha_{goal} = Q_{max}^{-1}(Q_{goal})$. The* efficiency *of α-allocation is measured by the distance between the superposed α-allocation, $\alpha_t(k) \triangleq \sum_{i=1}^{n} \alpha_i(k)$, and its target value α_{goal}.* ■

Neither over-allocation $\alpha_t(k) > \alpha_{goal}$, nor under-allocation $\alpha_t(k) < \alpha_{goal}$ is desirable and efficient, as over-allocation may result in packet losses and under-allocation yields poor transient response, buffer utilization, and transmission throughput. The goal of α-control is to drive the total or aggregate α-allocation $\alpha_t(k)$ of $\alpha(k)$ to α_{goal} as close and as fast as possible from any initial state.

Definition 2.6.2 *The* fairness *of α-allocation $\alpha(k) = (\alpha_1(k), \cdots, \alpha_n(k))$ for n multicast connections of the same priority sharing the common bottleneck at time k is measured by the* fairness index $\phi(\alpha(k)) \triangleq \frac{[\sum_{i=1}^{n} \alpha_i(k)]^2}{n\ [\sum_{i=1}^{n} \alpha_i^2(k)]}$. ■

Notice that $\frac{1}{n} \leq \phi(\alpha(k)) \leq 1$. $\phi(\alpha(k)) = 1$ if $\alpha_i(k) = \alpha_j(k)$, $\forall i \neq j$, corresponding to the "best" fairness. $\phi(\alpha(k)) = \frac{1}{n}$ if α is allocated to only one of n active connections.

This corresponds to the "worst" fairness and $\phi(\alpha(k)) \rightarrow 0$ as $n \rightarrow \infty$. So, the fairness index $\phi(\alpha(k))$ should converge as close to 1 as possible as $k \rightarrow \infty$.

The α-control is a negative feedback control over the rate-gain parameter, and computes $\alpha(k+1)$ based upon the current value $\alpha(k)$ and the feedback $BCN(k-1, k)$. Thus, $\alpha(k+1)$ can be expressed by the control function as $\alpha(k+1) = g(\alpha(k), BCN(k-1, k))$. For implementation simplicity, we only focus on a linear control function $g(\cdot, \cdot)$ by which we mean that $\alpha(k+1) = p + q\alpha(k)$, where coefficients p and q are determined by feedback information $BCN(k-1, k)$. The theorem given below describes the feasibility and optimality of the linear α-control, which ensures the convergence of α-control to the efficiency and fairness of buffer allocation as defined by Definitions 2.6.1 and 2.6.2.

Theorem 2.6.1 *Suppose n connections sharing a common bottleneck are synchronously flow-controlled by the proposed α-control. Then, (1) in* transient *state, the α-control law is feasible and optimal linear control in terms of convergence to the efficiency and fairness of buffer allocation; (2) in* equilibrium *state, the α-control law is feasible and optimal linear control in terms of maintaining the efficiency and fairness of buffer allocation.*

Proof. The detailed proof is provided in Appendix I. ∎

Remarks on Theorem 2.6.1: Theorem 2.6.1 is an extension from bandwidth control [22] to buffer control, but differs from [22] as follows. Unlike the bandwidth control exerted at the control-packet transmission rate, the α-control is exercised once every rate-control cycle. As a result, the α-control distinguishes transient state from equilibrium state, and applies different control algorithms in these two states, which makes $\alpha_t(k)$ not only monotonically converge to, but also lock within, a small neighborhood of its target α_{goal}. Since the total allocation $\alpha_t(k)$, or the number of connections, keeps on going up and down due to cross-traffic variations in real-world networks (or equivalently, the target α-allocation for each connection is "moving" up and down), it suffices to ensure convergence to fairness/efficiency in transient state and maintain the achieved fairness/efficiency in equilibrium state.

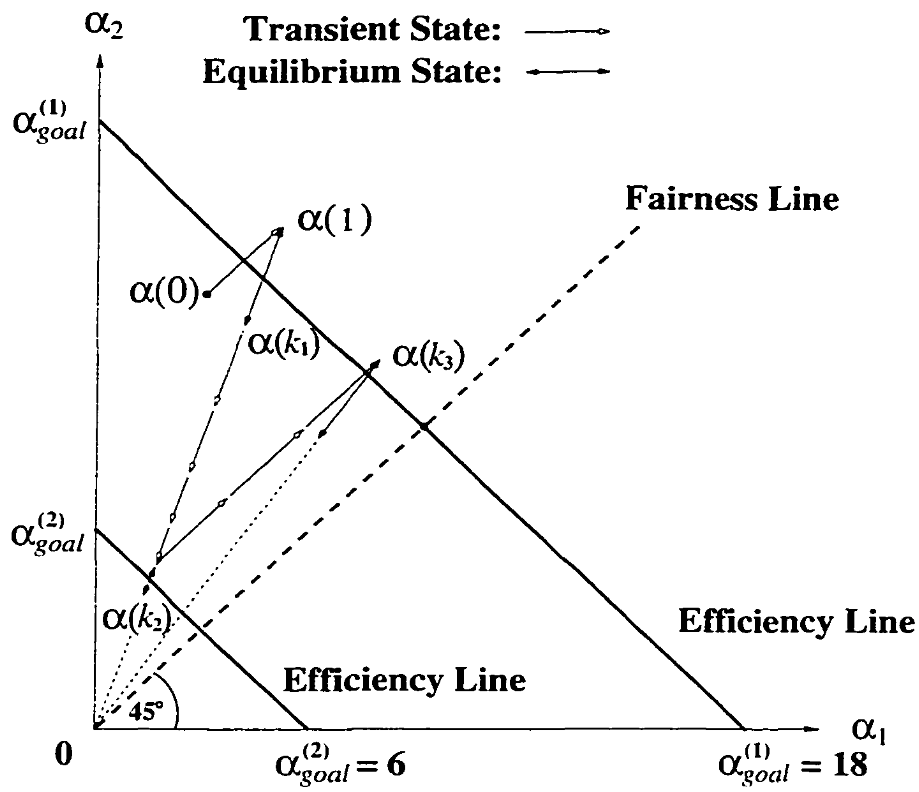


Figure 2.12: α-allocation convergence to efficiency and fairness: $\alpha(k) \longrightarrow$ efficiency/fairness.

Using the analysis of Section 2.3, we consider two examples given below in a 2-dimensional space (for two connections) to show the convergence of α-allocation under the α-control in terms of efficiency and fairness. As shown in Figures 2.12 and 2.12, any α-allocation of two connections at the k-th α-control is represented as a point $\alpha(k) = (\alpha_1(k), \alpha_2(k))$ in a 2-D space. All allocation points (α_1, α_2) for which $\alpha_1 + \alpha_2 = \alpha_{goal}$ form the *efficiency line*, and all points for which $\alpha_1 = \alpha_2$ form the *fairness line* which is a 45° line. It is easy to verify that an additive increase, $(\alpha_1, \alpha_2) + p \triangleq (\alpha_1 + p, \alpha_2 + p)$, corresponds to moving up ($p>0$) along the 45° line, and a multiplicative decrease or increase, $q(\alpha_1, \alpha_2) \triangleq (q\alpha_1, q\alpha_2)$ ($0<q<1$ or $q>1$), corresponds to moving along the line that connects the origin to (α_1, α_2).

EXAMPLE 1. Let two connections sharing a bottleneck be α-flow-controlled. The connection bottleneck is characterized by: $\mu = 184$ packets/ms, $Q_{goal} = 200$ packets, $Q_h = 18$

packets, and $\tau = 2$ ms (so, $\alpha_{goal} = 18$ packets/ms^2). Consider a scenario (see Fig. 2.12) where α_{goal} is equal to $\alpha^{(1)}_{goal} = 18$ initially, but reduces to $\alpha^{(2)}_{goal} = 6$ at the k_1-th α-control, and then returns to $\alpha^{(1)}_{goal}$ after the k_2-th α-control. The variation of α_{goal} is due to the variation in the number of connections between $n = 2$ and $n = 6$, or due to the variations in τ between $\tau^{(1)} = 2$ ms and $\tau^{(2)} = 3.34$ ms. We take $q = 0.8$ and $p = 4$ for the two connections with $Q_{goal} = 200$ and $\tau = 2$ ms. Thus, $\frac{1}{2}p = 2$ for each of the two connections. Suppose $\alpha(0) = (3.035, 12.76)$ initially. Then, by α-control, $\alpha(1) = \alpha(0) + 2 = (5.035, 14.76)$ and $\alpha(2) = 0.8\alpha(1) = (4.028, 11.81)$ since $\alpha_1(0) + \alpha_2(0) = 15.795 < \alpha^{(1)}_{goal}$ and $\alpha_1(1) + \alpha_2(1) = 19.795 > \alpha^{(1)}_{goal}$. Thus, α-control enters equilibrium state around $\alpha^{(1)}_{goal}$ during which $\alpha(k)$ fluctuates between $(4.028, 11.81)$ and $(5.035, 14.76)$. When α_{goal} reduces to $\alpha^{(2)}_{goal}$, equilibrium is broken and $\alpha(k)$ converges to a new equilibrium state multiplicatively in 5 α-control iterations, and fluctuates between $(1.32, 3.87)$ and $(1.65, 4.838)$. Finally, α_{goal} returns back to $\alpha^{(1)}_{goal}$, $\alpha(k)$ converges to the new equilibrium state additively through 3 α-control iterations and fluctuates between $(6.12, 8.671)$ and $(7.65, 10.838)$. We observe that in transient state, α-control not only guarantees the monotonic convergence to the neighborhood of efficiency-line in both increase and decrease phases, but also improves the fairness index from $\phi(\alpha(0)) = 0.725$ to $\phi(\alpha(k_3)) = 0.971$ as shown in Fig. 2.12, where $\alpha(k_3) = (7.65, 10.838)$ is closer to the fairness line than $\alpha(0) = (3.035, 12.76)$.

EXAMPLE 2. The second example compares α-control with the AIMD algorithm applied to α (see Fig. 2.13). The parameters and $\alpha(0)$ are the same as in EXAMPLE 1 except that α_{goal} reduces to, and stays with, $\alpha^{(2)}_{goal}$ after $\alpha(k)$ reaches $\alpha(1)$. We observe that both schemes share the control trajectory from $\alpha(0)$ to $\alpha(k_1)$. However, after $\alpha(k)$ is driven to $\alpha(k_1)$, the two trajectories split. Under the α-control, $\alpha(k)$ converges to an equilibrium state and locks itself within a small neighborhood of $\alpha^{(2)}_{goal}$: $\{(1.32, 3.87), (1.65, 4.838)\}$. In contrast, under the AIMD algorithm, $\alpha(k)$ does not confine itself within a small neighborhood of $\alpha^{(2)}_{goal}$ and, in fact, $\alpha(k)$ cannot even reach any equilibrium state. The resultant maximum buffer-allocation "overshoot" for the AIMD at $\alpha(k_2)$ is as high as $Q^{(k_2)}_{max} - Q_{goal}$

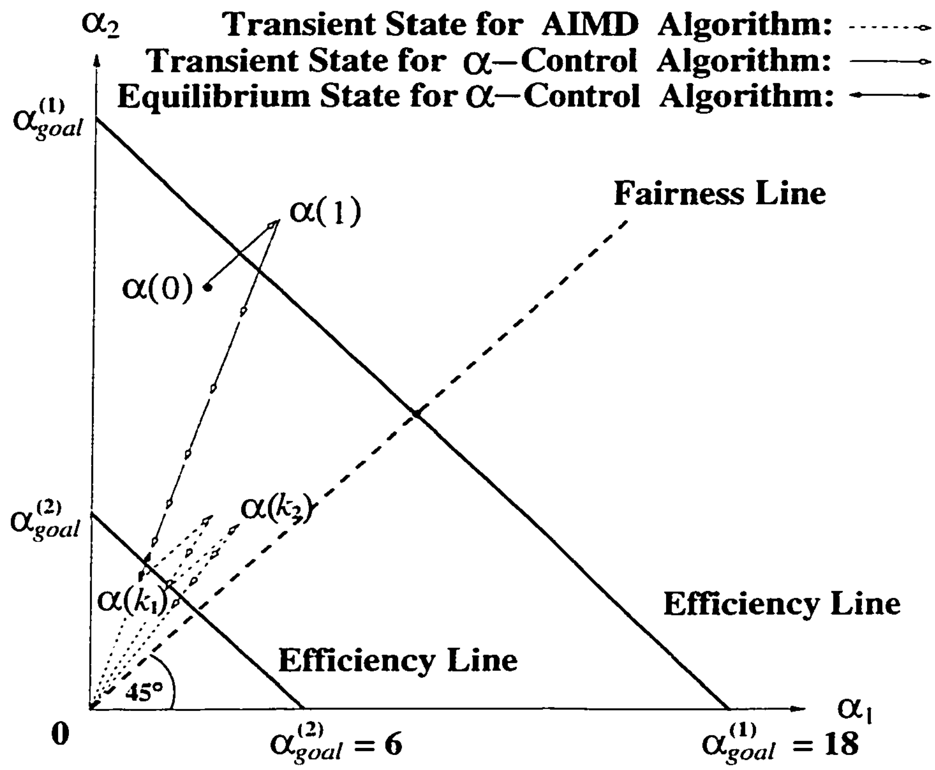


Figure 2.13: α-allocation convergence to efficiency and fairness: α-control vs. AIMD.

= 261 − 200 = 61 packets, which is about 9 times as large as that for α-control (with the maximum overshoot equal to $Q_{max}^{(k_1+1)} - Q_{goal} = 207 - 200{=}7$). So, even though the AIMD algorithm is better than α-control in term of speed of convergence to fairness, the AIMD's maximum buffer requirement and potential loss rate are much higher than α-control, especially when the variation in the number of connections or RTT is large.

2.6.2 Fluid Modeling and Analytical Results

M (> 1) concurrent flow-controlled connections with a common multicast-tree bottleneck are modeled by a single buffer and a server shared by M source rates $R_i(t)$. At time t the aggregate arrival rate at the multicast-tree bottleneck is $\sum_{i=1}^{M} R_i(t - T_f^{(i)})$. So, the bottleneck's queue length function at time t is

$$Q(t) = \begin{cases} 0; & \text{if } Q(t) = 0 \wedge \sum_{i=1}^{M} R_i(t) < \mu \\ \int_{t_0}^{t} \left\{ \sum_{i=1}^{M} R_i(v - T_f^{(i)}) - \mu \right\} dv + Q(t_0); & \text{if (1) } \sum_{i=1}^{M} R_i(t) > \mu; \text{ or} \\ & \text{(2) } \sum_{i=1}^{M} R_i(t) < \mu \wedge Q(t) > 0 \end{cases} \tag{2.48}$$

where $T_f^{(i)}$ is the forward delay for the i-th connection. Applying the same rate-control algorithm proposed in Section 2.2, for $i = 1, 2, \cdots, M$, we get:

$$R_i(t) \quad = \quad \begin{cases} R_i(t_0) + \alpha^{(i)}(t - t_0); & \text{if } Q(t - T_b^{(i)}) < Q_l \\ R_i(t_0) e^{-(1-\beta^{(i)})\frac{(t-t_0)}{\Delta_i}}; & \text{if } Q(t - T_b^{(i)}) \geq Q_h \end{cases} \tag{2.49}$$

The α-control is applied in the same way as in the single multicast connection case, but $Q_{max}^{(n)}$ is contributed, and Q_{goal} is shared, by all M connections.

Derivation of analytical results for multiple concurrent multicast connections is quite lengthy, thus omitted. Applying the derived analytical results to some simple multiple connection cases, we have already shown in [33] that the proposed scheme based on α-control is stable and efficient, and outperforms the schemes without α-control in dealing with RM-cell RTT and bandwidth variations, and achieving fairness in both buffer and bandwidth occupancies. Due to lack of space, we omit the analytical evaluation and refer the interested readers to [33] for more details. Instead, in the next section we present the simulation results to (1) verify the analytical results and (2) analyze the performance of the proposed scheme for more general cases where the locations, the number, and the bandwidth of multicast-tree bottlenecks vary with time.

2.6.3 Simulation Results

We conducted extensive simulations for concurrent multiple multicast VCs (Virtual Circuits) with multiple bottlenecks to study the performance of the proposed scheme with α-control, and compare it with schemes without α-control. By removing the assumptions

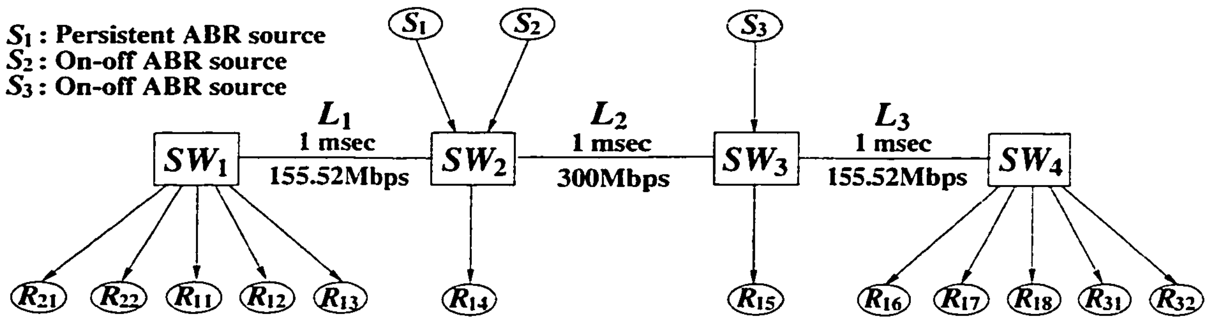
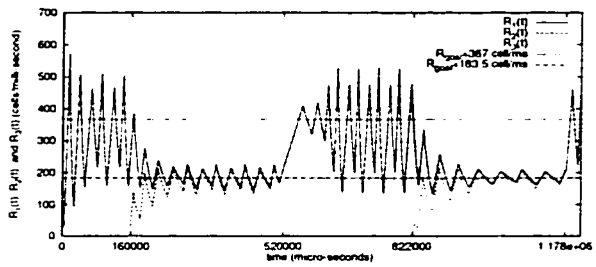


Figure 2.14: Simulation model for multiple multicast VCs.

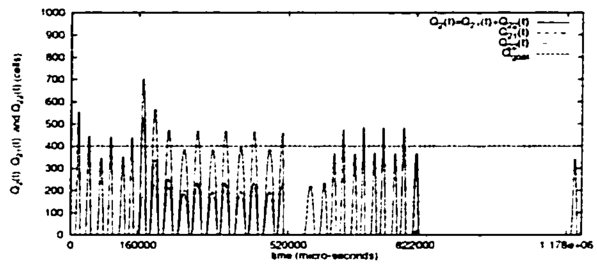
made for the modeling analysis, the simulation experiments accurately capture the dynamics of real networks, such as the noise-effect of RM-cell RTT due to the randomness of network environments, and RM-cell processing and queuing delays, instantaneous variations of bottleneck bandwidths, which are very difficult to deal with analytically.

The simulated network is shown in Figure 2.14, which consists of 3 multicast VCs running through 4 switches $SW_1, SW_2, \cdots, SW_4$ connected by 3 links L_1, L_2, L_3. S_i is the source of VC$_i$, $i = 1, 2, 3$, and R_{ij} is S_i's j-th receiver. So, VC$_2$ and VC$_3$ share L_1 and L_3, respectively, with VC$_1$. S_1 is a persistent ABR source which generates the main data traffic flow. S_2 and S_3 are two periodic on-off ABR sources with on-period = 360 ms and off-period = 1011 ms, respectively, which mimic cross-traffic noises, causing the bandwidth to vary dynamically at the bottlenecks. We set L_i's bandwidth capacity μ_i to (1) $\mu_1 = \mu_3$ = 155.52 Mbps and (2) μ_2 = 300 Mbps, forcing the potential bottlenecks L_1 and L_3 to show up. Letting all links' delays be 1 ms, S_1's RM-cell RTTs via R_{16}, R_{17}, R_{18} equal 4 ms which is 2 times of S_1's RM-cell RTTs via R_{11}, R_{12}, R_{13}.

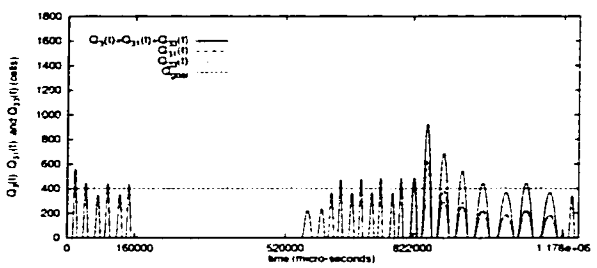
We implemented the simulation model by using the NetSim event-driven simulator [31]. The flow-control parameters used in the simulation remain the same as those used in the analytical solutions for comparison purposes. Specifically, Q_h = 50 cells, Q_{goal} = 400 cells, $\Delta = 0.4$ ms, $q = 0.6$, $p = 16.67$ cells/ms^2, and $R_0 = 30$ cells/ms; VC$_1$'s $\alpha_0 = 57.8$ cells/ms^2, VC$_2$ and VC$_3$'s $\alpha_0 = 22.9$ cells/ms^2. We let S_1 start at $t = 0$, S_2 at $t = 160$ ms, and S_3 at $t = 822$ ms such that S_2 and S_3 generate the cross-traffic noises against the main data



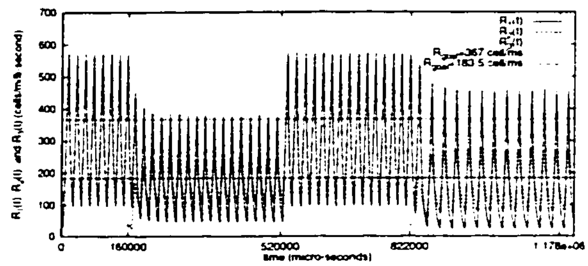
(a) $R(t) \rightarrow$ target bandwidth with α-control



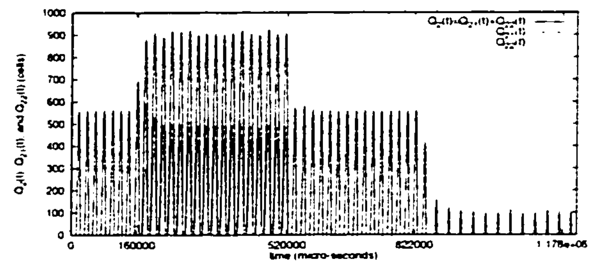
(b) SW_2: Total $Q_{max} \rightarrow Q_{goal}$ with α-control



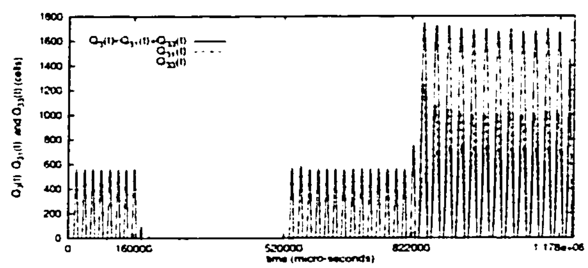
(c) SW_3: Total $Q_{max} \rightarrow Q_{goal}$ with α-control



(d) $R(t) \not\rightarrow$ target bandwidth without α-control



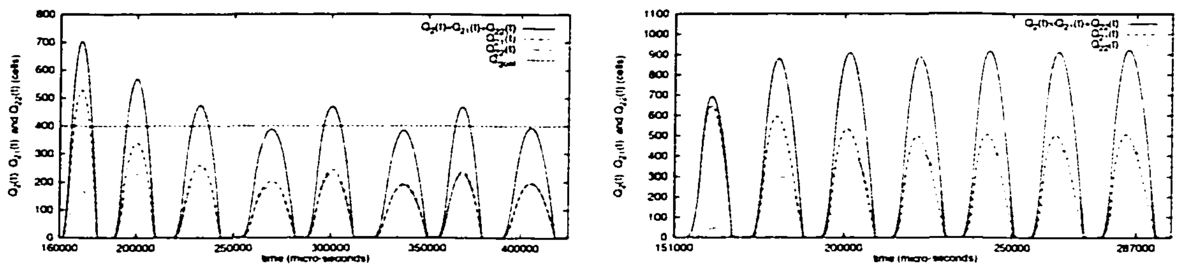
(e) SW_2: Total $Q_{max} \not\rightarrow Q_{goal}$ without α-control



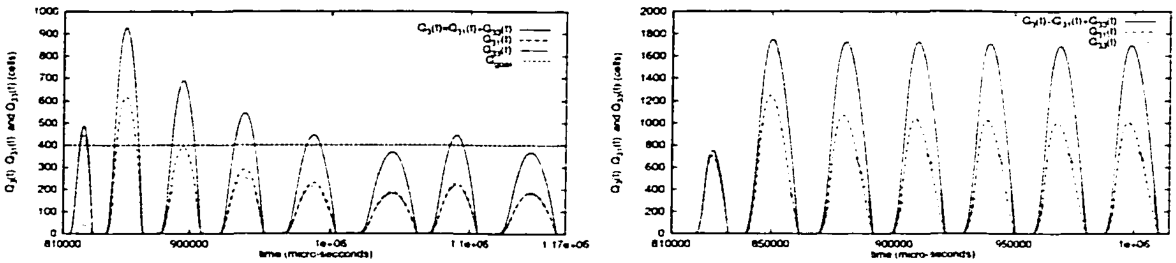
(f) SW_3: Total $Q_{max} \not\rightarrow Q_{goal}$ without α-control

Figure 2.15: Dynamics performance comparison between schemes with and without α-control.

traffic flow at the potential bottlenecks L_1 and L_3 with the respective on-periods appearing alternately without any overlap in time. Consequently, as shown in Figures 2.15(a)–(f), the first two on-periods of VC$_2$ and VC$_3$ divide the first 1178 ms simulation time axis into the following 4 time periods (ms). $T_1 = [0, 160]$ where only VC$_1$ is active; $T_2 = [160, 520]$ where both VC$_1$ and VC$_2$ are active; $T_3 = [520, 822]$ where only VC$_1$ is active; $T_4 = [822, 1178]$ where both VC$_1$ and VC$_3$ are active. The simulation results for the two different schemes are summarized in Figures 2.15(a)–(f) and Figures 2.16(a)–(d), where all results with α-control are plotted in Figures 2.15(a)–(c) and Figures 2.16(a)–(b) on the left, while those without



(a) SW_2: $Q_{21}(t)$, $Q_{22}(t)$ $\rightarrow$ fairness with α-control (c) SW_2: $Q_{21}(t)$, $Q_{22}(t)$ $\not\rightarrow$ fairness without α-control



(b) SW_3: $Q_{31}(t)$, $Q_{33}(t)$ $\rightarrow$ fairness with α-control (d) SW_3: $Q_{31}(t)$, $Q_{33}(t)$ $\not\rightarrow$ fairness without α-control

Figure 2.16: Buffer occupancy fairness comparison between schemes with and without α-control.

α-control are shown in Figures 2.15(d)–(f) and Figures 2.16(c)–(d) on the right. Each individual performance measure with α-control is compared with its counterpart without α-control listed in the same row.

(1) During T_1. For the α-controlled scheme, Figure 2.15(a) shows that VC_1's rate $R_1(t)$ converges to L_1 and L_3's capacity 367 cells/ms (155.52 Mbps) since VC_1 is the only active VC and it grabs all the bandwidth available. Thus, during T_1, there exist 2 bottlenecks located at L_1 and L_3 with RTT equal to 2 ms and 4 ms, respectively. Denote these two bottlenecks' total queue lengths at SW_2 and SW_3 by $Q_2(t)$ and $Q_3(t)$ and their maximum by $Q_{max}^{(2)}$ and $Q_{max}^{(3)}$, respectively. From Figures 2.15(a)–(c) we observe that after experiencing one transient cycle due to $Q_{max}^{(2)} = Q_{max}^{(3)} = 560 > Q_{goal}$, $Q_{max}^{(2)}$ and $Q_{max}^{(3)}$ converge to Q_{goal}'s neighborhood $[350, 446]$ by α-control. So, α-control not only drives $R_1(t)$ to its target bandwidth, but also confines the maximum queue lengths at the bottlenecks to Q_{goal}'s neighborhood. In contrast, for the schemes without α-control, Figures 2.15(d)–(f) show that $R_1(t)$ converges to $\mu_1 = \mu_3 = 367$, but $Q^{\langle 2\rangle}_{max} = Q^{\langle 3\rangle}_{max} = 560$ and never went down to $Q_{goal} = 400$.

(2) During T_2. VC_2 starts transmission, and competes for bandwidth and buffer space with VC_1. The bottleneck at L_3 is expected to disappear since $R_1(t)$'s new target bandwidth along path via L_1 is only a half of that via L_3. So, L_1 is the only bottleneck with RTT = 2 ms, target bandwidth = $\frac{1}{2}\mu_1$ for each of VC_1 and VC_2. For the α-controlled scheme, Figure 2.15(a) shows that the source rates $R_1(t)$ and $R_2(t)$ experience two transient cycles during which $R_1(t)$ gives up $\frac{1}{2}\mu_1$ to $R_2(t)$ until they reach a new equilibrium. Figure 2.15(b) shows that a large queue build-up $Q^{\langle 2\rangle}_{max} = 704$ as a result of the superposed rate-gain parameter from $R_1(t)$ and $R_2(t)$, and the reduced bottleneck bandwidth. With α-control, $Q^{\langle 2\rangle}_{max}$ is driven down to Q_{goal}'s neighborhood of $[385, 468]$. Figure 2.15(c) shows $Q_3(t) = 0$, verifying that the bottleneck at L_3 vanished. Figure 2.16(a) is a zoom-in picture of $Q_2(t) = Q_{21}(t) + Q_{22}(t)$ of Figure 2.15(b), where $Q_{21}(t)$ is the per-VC queue of VC_1 and $Q_{22}(t)$ is the per-VC queue of VC_2 at SW_2, respectively. Figure 2.16(a) indicates that in the first transient cycle, $Q_{21}(t)$'s maximum $Q^{\langle 21\rangle}_{max} = 528$, which is more than 3 times of $Q_{22}(t)$'s maximum $Q^{\langle 22\rangle}_{max} = 175$. Under α-control, $Q_{21}(t)$ and $Q_{22}(t)$ converge to each other quickly and become identical from $t = 391$ ms. This verifies that the α-control law can ensure the fairness in buffer occupancy between the competing VCs. By contrast, for the scheme without α-control, Figure 2.15(e) illustrates that $Q^{\langle 2\rangle}_{max}$ jumps up to as high as 900 and stays at 900 even after the transient state. Figure 2.16(c), the zoom-in picture of Figure 2.15(e), shows that $Q_{21}(t)$ never converges to $Q_{22}(t)$ even after the transient state, and thus the buffer space is not fairly occupied.

(3) During T_3. After VC_2 goes into an off-period, $R_1(t)$ grabs all the bandwidth of μ_1 again. After $R_1(t)$ reaches the L_1's bandwidth capacity, the bottleneck at L_3 also shows up due to $\mu_1 = \mu_3$, and then the total number of bottlenecks becomes 2 again. For the scheme with α-control, because $Q_{22}(t)$ suddenly drops to zero as VC_2 goes into an off-period, making $Q^{\langle 2\rangle}_{max} \ll Q_{goal}$, which generates 3 consecutive $BCI = 0$, the α-

control's additive-increase operation $\alpha_n = \alpha_{n-1} + p$ is executed twice during the transient cycles until $Q_{max}^{\langle 2 \rangle}$ converges to Q_{goal}'s neighborhood $[367, 483]$ within 3 transient cycles. Note that $Q_{max}^{\langle 2 \rangle}$ *monotonically* converges to $[367, 483]$ as shown in Figure 2.15(b). This is expected since $p = 16.67 \le \left(\frac{1-q}{q}\right)\left(\frac{\sqrt{Q_{goal}} - \sqrt{2Q_h}}{\tau}\right)^2$, satisfying the condition (3) in Theorem 2.4.3. This observation further verifies the correctness of the optimal monotonic convergence condition derived in Theorem 2.4.3. In Figures 2.15(d)–(e) for schemes without α-control, the queue and rate dynamics simply repeat their dynamics in T_1, suffering from a large buffer requirement.

(4) During T_4. The rate and queue dynamics are similar to T_2's, except that the bottleneck is now located at L_3 with a new target bandwidth $= \frac{1}{2}\mu_3$ and a longer RTT $= 4$ ms. For the α-controlled scheme, Figure 2.15(b) shows $Q_2(t) = 0$, indicating that the bottleneck at L_1 disappeared and L_3 is the only bottleneck. Figure 2.15(c) shows that $Q_{max}^{\langle 3 \rangle}$ shoots up to 928, as a result of the doubled RTT (4 ms) via L_3. Within 3 transient cycles, $Q_{max}^{\langle 3 \rangle}$ converges to Q_{goal}'s neighborhood of $[367, 445]$ in equilibrium state. Figure 2.16(b), a zoom-in picture of Figure 2.15(c), shows the buffer-occupancy fairness ensured by α-control. These observations verify that α-control can efficiently adapt to RM-cell RTT variations in terms of buffer requirement and fairness. By contrast, for the scheme without α-control, Figures 2.15(e)–(f) show 2 bottlenecks: (1) a bandwidth-congestion bottleneck at L_1; (2) a buffer-congestion bottleneck at L_3. Figure 2.15(f) shows that $Q_{max}^{\langle 3 \rangle} = 1740$, almost 2 times of that under the α-controlled scheme. More importantly, $Q_{max}^{\langle 3 \rangle}$ stays around 1740 even after the transient state. Moreover, Figure 2.16(d), a zoom-in picture of Figure 2.15(f), demonstrates that buffer occupancy is not fair because $Q_{max}^{\langle 31 \rangle} = 1000$ but $Q_{max}^{\langle 33 \rangle} = 740$.

The three VCs' average throughputs $\overline{R}_1, \overline{R}_2, \overline{R}_3$ (for on-off sources averaging over the on-period only) obtained by the simulation are compared for the two types of schemes in Table 2.1. In all the three VC cases the proposed scheme with α-control is observed to outperform the scheme without α-control in terms of average throughput.

scheme type	$\overline{R}_1$ of VC_1	$\overline{R}_2$ of VC_2	$\overline{R}_3$ of VC_3
with α-control	234.448	150.671	147.709
without α-control	209.367	143.672	137.655

Table 2.1: Average throughputs (cells/ms) of schemes with and without α-control.

2.7 Conclusion

We proposed and analyzed a flow-control scheme for multicast ATM ABR services, which scales well and is efficient in dealing with the variations in the multicast-tree structure and RM-cell RTT. We identified the main features of multicast ABR flow control and incorporated them into the design of our control algorithm. We developed the α-control, the second-order rate control, algorithm to handle the variation of RM-cell RTT. By exercising two-dimensional rate control, the proposed scheme not only makes the transmission rate converge to the available bandwidth of the multicast connection's most congested branch path sensed/perceived by the source, but also brings the buffer occupancy to a small neighborhood of the target setpoint bounded by buffer capacity. By employing a "soft" feedback synchronization mechanism, the proposed scheme scales well with the size of multicast tree. Non-responsive branches are also detected quickly with non-responsive timers and connection-update vectors.

Applying the fluid analysis, we modeled the proposed flow-control scheme and analyzed the system dynamic behavior for multicast ABR services under the persistent traffic sources. We derived closed-form expressions for queue buildup, average throughput, and other flow-control measures in both transient and equilibrium states. These expressions were then used to evaluate the system performance and studies the α-control's convergence properties. We derived an analytical relationship between the rate-gain parameter and RM-cell RTT subject to both cell-lossless transmission and finite buffer capacity constraints. This analytical relationship ensures the feasibility of the α-control in dealing with RM-cell RTT variations and provides an insight on how the required buffer space can be controlled by adjusting the rate-gain parameter. We developed an optimal control condition, under which the α-control guarantees the monotonic convergence of system state to the optimal regime from an arbitrary initial value. We also derived the closed-form expressions that upper-bound the size of convergence regime in the buffer requirements under the optimal control condition.

Analytical results show our scheme based on α-control to be stable and efficient in that both the source rate and bottleneck queue length rapidly converge to a small neighborhood of the designated operating point. The dynamic performance of the proposed scheme with a single multicast connection is quantitatively evaluated by both modeling analysis and simulation experiments. The simulation results verified the analytical results in both transient and equilibrium states. We proved that α-control guarantees the fairness of buffer utilization among multiple concurrent multicast connections. We also derived the greatest lower bound of target buffer occupancy to determine the optimal α-control parameters. We carried out loss control analysis, demonstrating the feasibility and effectiveness of the α-control in improving bandwidth utilization.

To accurately capture the dynamics of real networks, the extensive simulation experiments were conducted for concurrent multiple multicast-connections where the number, location, and bandwidth of bottlenecks vary with time. The simulation experiments for multiple multicast connections demonstrate the superiority of the proposed scheme to the other schemes in dealing with the variations of RM-cell RTT and link bandwidth, achieving fairness in both buffer and bandwidth occupancies, and increasing average throughput. Our ongoing work focuses on the error-control algorithms and performance for multicast ABR services and the extension of the proposed scheme to the ER-based flow-control schemes.

CHAPTER 3

MULTICAST SIGNALING PROTOCOLS AND ITS DETERMINISTIC DELAY MODELING

3.1 Introduction

A flow-control algorithm consists of two fundamental components: rate control and flow-control signaling. These two components are conceptually separate from the flow-control theory standpoint, but are often blended together in most flow-control algorithms. Rate control adapts the source rate to the dynamic variation of available network bandwidth. Flow-control signaling delivers the information related to congestion and rate-control between the source and network/receivers. Consequently, this signaling is critically important to flow control because the source relies solely on the signaling information in making *correct* and *timely* flow-control decisions. Designing an efficient flow-control signaling protocol is difficult because the signaling messages — unlike data or audio/video traffic — tolerate neither error nor a large latency. A signaling message could be useless or even harmful if it is not accurate and its delay is very large. In other words, signaling traffic must meet both timeliness and reliability requirements. In ATM ABR services, flow-control signaling relies on the RM (Resource Management) cells, which convey the rate-control and congestion information among the source rate-controller, network switches, and the receiver.

Signaling for multicast flow control introduces two additional problems: *scalability* and *feedback-synchronization*. These two problems are closely related in the signaling protocol for multicast flow control. First, simultaneous feedback arrivals from all downstream branches can cause a *feedback implosion* [13] at the source or at a branch point, especially when the multicast tree is large. Hence, it is important for each branch point to consolidate the congestion-information feedback from its downstream branches and send only the consolidated feedback to its upstream node. Second, we need a feedback-synchronization signaling algorithm for consolidating feedbacks at each branch point, because different downstream branches' feedbacks may arrive at significantly different times.

The first-generation feedback consolidation algorithms [16–19, 34] for multicast ABR flow control employ a simple hop-by-hop (HBH) mechanism to deal with the feedback-implosion problem. On receipt of one forward RM cell at each branch node, only *one* consolidated feedback RM cell is propagated upward by a *single* hop. While this HBH scheme ensures that at each node of the multicast tree, the ratio of feedback RM cells to the forward RM cells is not larger than 1, it does not scale well with the multicast-tree topology because the RM cell round-trip time (RTT) is proportional to the height of the multicast tree. The multicast signaling performance would become unacceptably poor if the delay of a signaling message increases with the multicast-tree height. Thus, the HBH scheme does not scale well with respect to signaling delay. Moreover, since the feedback RM cells from downstream nodes are *randomly* consolidated without strict synchronization (or *freely-synchronized*) at branch points, the source may be misled by this incomplete feedback information, which can cause the *consolidation noise* problem [15, 20]. So, the HBH scheme performs poorly in the sense of signaling accuracy.

To reduce the RM-cell RTT and improve the multicast signaling accuracy, the authors of [15, 20] proposed feedback synchronization by accumulating feedback from *all* branches. The main drawback of this scheme is its slow transient response, as the feedback from the congested branch may have to needlessly wait for the feedback from longer paths, which may not be congested at all. The authors of [21] proposed an algorithm to speed up the transient response by sending fast congestion feedback without waiting for all branches' feedback during the transient phase.

One critical deficiency of the schemes described above is that they do not detect and remove non-responsive branches during feedback synchronization. One or more non-responsive branches may detrimentally impact signaling accuracy and timeliness by providing either stale congestion information, or by stalling the entire multicast connection. In [6], we proposed a novel feedback-synchronization signaling algorithm, called the *Soft-Synchronization Protocol* (SSP), which derives a single consolidated RM cell at each branch point from feedback RM cells of different downstream branches that are not necessarily responses to the same forward RM cell in each synchronization cycle. The SSP not only scales well with the multicast-tree topology, but also can readily detect and remove non-responsive branches.

All of the above-referenced work only focused on the design and implementation of feedback-synchronization signaling algorithms. However, the delay properties of these algorithms are, despite their vital importance, neither well understood nor thoroughly studied. In this chapter, we introduce our proposed SSP in details for multicast signaling and develop balanced and unbalanced binary-tree models to characterize the delay performance of a class of feedback-synchronization signaling algorithms in terms of RM-cell RTTs. The benefits of these modeling and evaluation techniques presented in this chapter are two-fold. First, it enables a direct quantitative comparison between the SSP and HBH schemes. Using the deterministic binary-tree model, we derive the closed-form equations by which we can calculate each path's multicast signaling delay in any given multicast tree. We conduct numerical analysis which show that SSP outperforms HBH in terms of feedback-synchronization signaling delay in both cases of balanced and unbalanced multicast trees. Our analytical results also reveal that SSP can not only support efficient feedback-synchronization signaling, but also make the effective RM-cell RTT virtually independent of the multicast-tree's height and path-length variations. Second, the proposed modeling technique establishes a general signaling-delay evaluation framework for all feedback-synchronization algorithms. While our evaluation focuses on the signaling delay for ABR multicast flow-control in ATM networks, the modeling technique is not confined to an ABR multicast environment, and can be applied to the signaling delay analysis for *any* feedback-synchronization based multicast algorithm.

This chapter is organized as follows. Section 3.2 presents an overview of SSP for completeness. In Section 3.3, we introduce the binary-tree model, and apply it to analytically derive the signaling delay properties of each path for both the SSP and HBH schemes. Section 4.2 studies the statistical properties of multicast signaling delay in terms of average multicast-tree RM-cell RTTs and delay variations for both the SSP and HBH schemes. Section 4.6 describes the simulation results, verifying the analytical results. In Section 3.5, we derive the optimal RM-cell interval for SSP to minimize the RM-cell RTTs for a given multicast tree. The chapter concludes with Section 3.6.

3.2 Description of SSP

We first present an overview of SSP, especially the switch feedback-synchronization algorithm [6, 11]. At the heart of SSP is a pair of connection-update vectors: (i) *conn_patt_vec*, the connection pattern vector where *conn_patt_vec*$(i) = 0$ (1) indicates the i-th output port of the switch is (not) a downstream branch of the multicast connection. Thus, *conn_patt_vec*$(i) = 0$ (1) implies that a data copy should (not) be sent to the i-th downstream branch and a feedback RM cell is (not) expected from the i-th downstream branch;[1] (ii) *resp_branch_vec*, the responsive branch vector is initialized to $\underline{0}$ and reset to $\underline{0}$ whenever a consolidated RM cell is sent upward from the switch. *resp_branch_vec*(i) is set to 1 if a feedback RM cell is received from the i-th downstream branch. The connection pattern specified in *conn_patt_vec* is updated by *resp_branch_vec* each time when the non-responsive branch is detected or a new connection request is received from a downstream branch.

A simplified pseudo-code of the switch RM-cell processing algorithm is given in Fig-

[1] Note that the negative logic is used for convenience of implementation.

```
On receipt of a feedback RM cell from the i-th branch:
if (conn_patt_vec(i) ≠ 1) { ! Only process connected branches;
  resp_branch_vec(i) := 1; ! Mark connected and responsive branch;
  MCI := MCI ∨ CI; ! Bandwidth-congestion indicator processing;
  MER := min{MER, ER}; ! ER information processing;
  if (conn_patt_vec ⊕ resp_branch_vec = 1) { ! soft feedback-synchronization;
    send RM cell (dir := back, ER := MER, CI := MCI); ! Send fully-consolidated RM cell upstream
    no_resp_timer := N_nrt;  ! Reset non-responsive timer;
    resp_branch_vec := 0);  ! Reset responsive branch vector;
    MCI := 0;  MER := ER;}}; ! Reset RM-cell control variables
On receipt of a forward RM cell:
multicast RM cell based on conn_patt_vec; ! Multicast RM cell to downstream branches
no_resp_timer := no_resp_timer − 1; ! No-responsive branch checking
if (no_resp_timer = 0) { ! There is a non-responsive branch;
  conn_patt_vec := resp_branch_vec ⊕ 1; ! update connection pattern vector;
  if (resp_branch_vec ≠ 0) { ! There is at least one responsive branch;
    send RM cell (dir := back, ER := MER, CI := MCI); ! Send partially-consolidated RM cell upstream;
    no_resp_timer := N_nrt; ! Reset non-responsive timer;
    resp_branch_vec := 0; ! Reset responsive branch vector;
    MCI := 0;  MER := ER; }}; ! Reset RM-cell control variables.
```

Figure 3.1: Pseudocode for switch feedback-synchronization algorithm.

ure 3.1. On receipt of a feedback RM cell from a connected downstream branch, the switch first marks its corresponding bit in *resp_branch_vec* and then conducts RM-cell consolidation operations. If the modulo-2 addition (the soft-synchronization operation), $conn\_patt\_vec \oplus resp\_branch\_vec = \underline{1}$, an all 1's vector, indicating all feedback RM cells are synchronized, then a fully-consolidated feedback RM cell is generated and sent upward. But, if the modulo-2 addition is not equal to $\underline{1}$, the switch needs to await other feedback RM cells for synchronization. Notice that since the synchronization algorithm allows feedback RM cells corresponding to different forward RM cells to be consolidated, the feedback RM cells are "softly-synchronized" or "loosely-synchronized" at branch nodes.

Upon receiving a forward RM cell, the switch first multicasts it to all the connected branches specified by *conn_patt_vec*. Then, it decrements the non-responsive timer for this connection by one. The *no_resp_timer* is initialized to a threshold N_{nrt}, and reset to N_{nrt} whenever a consolidated RM cell is sent upward. The pre-determined timeout value N_{nrt} for non-responsiveness is determined by such factors as the difference between the maximum and minimum RM-cell RTTs in a multicast tree. We use the forward RM-cell arrival time as a natural clock for detecting/removing non-responsive branches (such that it will still work even in the presence of faults in the downstream branches). Each time a switch receives a forward RM cell, the multicast connection's *no_resp_timer* is decremented by one. If $no\_resp\_timer = 0$ (timeout) and $resp\_branch\_vec \neq \underline{0}$ (i.e., there is at least one downstream branch responsive), then the switch will stop awaiting arrival of feedback RM cells and immediately generate a partially-consolidated RM cell, then send it upward. Whenever $no\_resp\_timer = 0$, at least one non-responsive downstream branch is detected and will be removed by the simple complementary operation: $conn\_patt\_vec := resp\_branch\_vec \ominus \underline{1}$, which updates *conn_patt_vec*. Thus, a downstream branch which has not sent any feedback RM cell for N_{nrt} forward RM-cell time units will be removed from the multicast tree.

3.3 The Deterministic Model of Multicast Signaling Delay

It is well-known that the feedback delay plays a crucial role in determining the effectiveness of any flow-control scheme [6]. In this section, we analyze the properties of RM-cell RTTs of each path for different feeback-synchronization algorithms.

3.3.1 The Binary-Tree Model

To simplify the analysis of RM-cell RTTs, we *quantize* the network feedback delay by assuming each switch-hop to have a uniform delay (including the processing and propagation delays). This assumption can be relaxed easily because the difference in switch-processing delays and the link-propagation delays of different switch-hops can be translated into different numbers of switch-hops, each with the same delay. We use the hop-delay, τ_h, which is the sum of the switch-processing delay and link-propagation delay taken in each hop, as the time unit in our delay analysis. To study the worst case and enable performance comparison, we only consider two types of multicast trees: *balanced* and *unbalanced* binary trees. Since we are only concerned with a path's RM-cell RTT which is determined by its length, it suffices to consider binary trees. Notice that in an unbalanced binary tree, the number of paths, denoted by n, from the root to all leaves equals the height of the tree, denoted by m, while in a balanced binary tree $n = 2^{m-1}$. Figure 3.2 illustrates these two types of trees with height $m = 4$.

As discussed in [6, 21], for ABR services only the feedback from the most-congested path in a mulicast tree governs the flow-control operations at the source. However, the RM-cell RTT of different paths in a mulitcast tree may vary sinificantly due mainly to the difference in their length. Thus, we need to analyze each individual path's RM-cell RTT in a multicst tree. The individual path's RTT is also affected by the feedback-synchronization algorithms used. In addition, the RM-cell RTT for a given path may vary at the beginning of the flow-control operation (in an initial state) when feedback RM cells are not yet "regularly" synchronized. The RM-cell RTT becomes stable after feedback RM cells are regularly synchronized (in a steady state). In what follows, we analyze the feedback-delay properties, in both initial and steady states, of each path in a multicast tree which is flow-controlled by the HBH and SSP schemes, respectively.

3.4 Multicast Signaling Delay Analysis on Each Path in a Multicast Tree

3.4.1 Feedback-Delay Properties for the HBH Scheme

The following theorem gives a set of formulas for calculating all paths' RM-cell RTTs in an unbalanced tree for the HBH scheme.

Theorem 3.4.1 *If an unbalanced multicast tree of height $m \geq 2$ is flow-controlled by HBH*

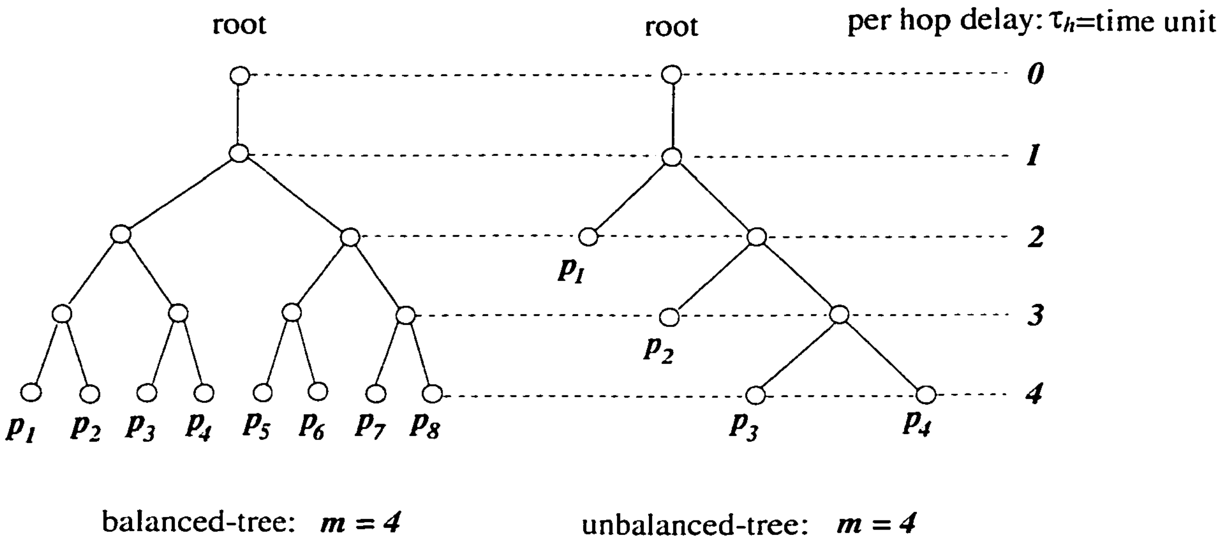


Figure 3.2: Balanced and unbalanced binary multicast trees.

with an RM-cell interval $\Delta \geq 1$ (τ_h), *then the RM-cell RTT, denoted by* $\tau_u(j, \Delta)$, *of the* j*-th (counting from left to right) path,* P_j, *remains the same in both steady and initial states, and is determined by:*

$$\tau_u(j, \Delta) = 2 + j\ \Theta(\Delta), \tag{3.1}$$

where $1 \leq j \leq m - 1$ *and* $\Theta(\Delta)$ *is the threshold function defined by*

$$\Theta(\Delta) \triangleq \max\{2, \Delta\} = \begin{cases} \Delta, & \text{if } 2 \leq \Delta \leq \tau_{max}; \\ 2, & \text{if } \Delta = 1; \end{cases} \tag{3.2}$$

where $1 \leq \Delta \leq \tau_{max}$,[2] $\tau_{max} = 2m$.

Proof. The proof is provided in Appendix J. ∎

The following corollary, providing equations to compute all paths' RM-cell RTTs in a balanced tree for the HBH scheme, is the direct result from Theorem 3.4.1.

Corollary 3.4.1 *If a balanced multicast tree of height* $m \geq 2$ *is flow-controlled by HBH with* $\Delta \geq 1$, *then RM-cell RTTs of all paths, denoted by* $\tau_b(j, \Delta)$, *are the same in both*

[2] Theorem 3.4.1 still holds even when $\Delta \geq \tau_{max} = 2m$. But the RM-cell update interval Δ is usually a fraction of the maximum RM-cell RTT. So, we do not consider the case of $\Delta \geq \tau_{max} = 2m$.

steady and initial states, and are determined by:

$$\tau_b(j, \Delta) = \max_{j \in \{1,2,\cdots,m-1\}} \{\tau_u(j, \Delta)\} = \tau_{max} + (m-1)[\Theta(\Delta) - 2]; \tag{3.3}$$

where $\tau_{max} = 2m$, $1 \leq j \leq 2^{m-1}$, *and* $\tau_u(j, \Delta)$ *and* $\Theta(\Delta)$ *are defined by Eqs. (3.1) and (3.2), respectively, for an unbalanced multicast tree of the same height.*

Proof. The proof follows by letting $j = m - 1$ in Eq. (3.1) of Theorem 3.4.1. ■

3.4.2 Feedback-Delay Properties for the SSP Scheme

The following lemma characterizes the synchronization relationships between paths under SSP, which lays the foundation for Lemma 3.4.2.

Lemma 3.4.1 *Consider an unbalanced multicast tree of height* $m > 2$. *Let* P_i *be a relatively shorter path than another path* $P_{\tilde{i}}$ *such that* $1 \leq i < \tilde{i} \leq m - 1$. *If the multicast tree is flow-controlled by SSP with RM-cell interval* $\Delta \geq 1$, *then* $P_{\tilde{i}}$*'s feedback RM cell need not wait for* P_i*'s feedback RM cell for synchronization at any branch node.*

Proof. The proof is provided in Appendix K. ■

The lemma given below reveals four *iff* conditions for a path's RM-cell RTT to attain its limiting minimum, which consists of propagation and processing delays only (i.e., no synchronization delay).

Lemma 3.4.2 *Let* P_j *be the* j*-th path in an unbalanced tree as defined in Lemma 3.4.1 with* $1 \leq j \leq m - 1$. *Then, the following four claims are equivalent for the steady-state RM-cell RTT:*

<u>Claim 1</u>: P_j*'s feedback RM cell need not wait for a longer path* $P_{\tilde{j}}$*'s* $(\tilde{j} > j)$ *feedback RM cell to achieve feedback synchronization at the first branch node from* P_j*'s leaf;*

__Claim 2__: *P_j's feedback RM cell need not wait for feedback RM cells for synchronization at* **any** *branch node on P_j;*

__Claim 3__: $\exists\, k \in \{0, 1, 2, \cdots\}$ *such that* $2(m-j-1)-k\Delta = 0$, *where* $1 \le j \le m-1$ *and* $1 \le \Delta \le \tau_{max} = 2m$;

__Claim 4__: *P_j's steady-state RM cell RTT $\tau_u(j,\Delta)$ attains its minimum and is given by:*

$$\tau_u(j,\Delta) = \min_{\Delta}\{\tau_u(j,\Delta)\} = 2(j+1) \tag{3.4}$$

where $1 \le j \le m-1$ *and* $1 \le \Delta \le \tau_{max} = 2m$.

Proof. The proof is provided in Appendix L. ■

Using Lemmas 3.4.1 and 3.4.2, we obtain the following theorem, which gives a set of formulas to calculate all paths' RM-cell RTTs during both initial and steady states in an unbalanced tree under SSP.

Theorem 3.4.2 *Let P_j be the j-th path of an unbalanced tree as defined in Lemma 3.4.1* $(1 \le j \le m-1)$. *If the multicast tree is flow-controlled by SSP with the RM-cell update interval* Δ $(1 \le \Delta \le \tau_{max} = 2m)$,[3] *then the following claims hold for* $j = 1, 2, \cdots, m-1$; $\tau_{max} = 2m$; $1 \le \Delta \le \tau_{max}$:

__Claim 1__: *The number of P_j's feedback RM cells going through initial state is determined by:*

$$k_j^* \triangleq \max_{k \in \{0,1,2,\cdots\}} \{k \mid 2(m-j-1) - k\Delta \ge 0\}; \tag{3.5}$$

__Claim 2__: *P_j's RM-cell RTT in steady state is determined by:*

$$\tau_u(j,\Delta) = \tau_{max} - k_j^*\Delta; \tag{3.6}$$

[3] Theorem 3.4.2 still holds for $\Delta > \tau_{max} = 2m$, but Δ is typically a fraction of the maximum RM-cell RTT $\tau_{max} = 2m$.

<u>Claim 3</u>: *The i-th RM-cell RTT during P_j's initial state is determined by:*

$$\tau_u(j,\Delta,i) = \begin{cases} \tau_{max} - (i-1)\Delta; & \text{if } k_j^* \geq 1 \wedge 1 \leq i \leq k_j^* \\ \tau_u(j,\Delta); & \text{if } k_j^* \geq 1 \wedge i > k_j^* \\ \tau_{max}; & \text{if } k_j^* = 0. \end{cases}$$

Proof. The proof is provided in Appendix M. ∎

The corollary described below, giving the equations for calculating all paths' RM-cell RTTs in a balanced tree under SSP, follows directly from Theorem 3.4.2.

Corollary 3.4.2 *If a balanced-tree multicast connection of height $m \geq 2$ is flow-controlled by SSP with the RM-cell interval $\Delta \geq 1$, then all paths' RM-cell RTTs, $\tau_b(j,\Delta)$, are the same in both steady and initial states and are determined by:*

$$\tau_b(j,\Delta) = \max_{j \in \{1,2,\cdots,m-1\}} \{\tau_u(j,\Delta)\} = \tau_{max} \tag{3.7}$$

where $\tau_{max} = 2m$, $1 \leq j \leq 2^{m-1}$, and $\tau_u(j,\Delta)$ given by Eq. (3.6) is P_j's RM-cell RTT for an unbalanced multicast tree of the same height.

Proof. The proof follows by letting $j = m-1$ in Eq. (3.5), which leads to $k_{m-1}^* = 0$ and thus $\tau_b(j,\Delta) = \tau_u(m-1,\Delta) = \tau_{max}$ by Eq. (3.6). ∎

Remarks on Theorem 3.4.1 and Theorem 3.4.2: Comparing Theorem 3.4.1 and Theorem 3.4.2, we make the following observations.

R1. For the HBH scheme, RM-cell RTT in initial state is the same as that in steady state. In contrast, for the SSP scheme, RM-cell RTT in initial state, if any, is larger than, and lower-bounded by, RM-cell RTT in steady state. For SSP, the initial state acts like a "warm-up" period for feedback RM cells to be synchronized at each branch node, during which the initial-state RM-cell RTTs converge to their corresponding steady-state values. The "warm-up" periods for P_j $(1 \leq j \leq m-1)$ are determined by the values of k_j^* given in Eq. (3.5).

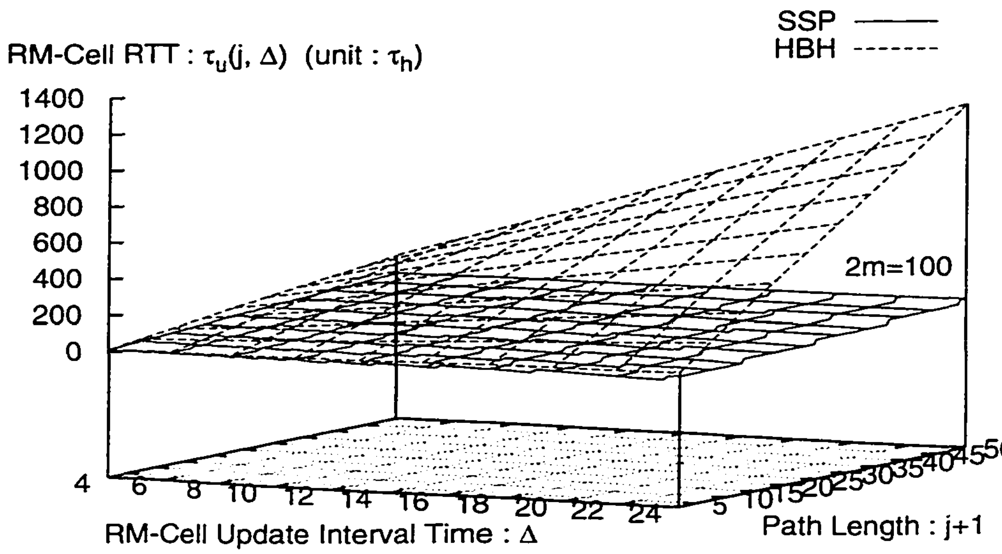


Figure 3.3: Impact of P_j's path length $j+1$, tree height m, RM-cell interval Δ on P_j's RM-cell RTT $\tau_u(j,\Delta)$: $\tau_u(j,\Delta)$ vs. $(j+1,\Delta)$, $(m=50)$.

R2. For SSP in both initial and steady states, the RM-cell RTT $\tau_u(j,\Delta)$ is upper-bounded by $\tau_{max} = 2m$ (see Claim 2 and Claim 3 of Theorem 3.4.2 and Eq. (3.7)). The increase rate of $\tau_u(j,\Delta)$ is $O(m)$ in the worst case. In contrast, for the HBH scheme, the RM-cell RTT $\tau_u(j,\Delta)$ is not upper-bounded by $\tau_{max} = 2m$ (see Eqs. (3.1) and (3.3)). Also, $\tau_u(j,\Delta)$ is very sensitive to path length $j+1$ and RM-cell update interval Δ, and increases at a rate up to $O(m^2)$ in the worst case.

3.4.3 Numerical Comparison of SSP and HBH

We present the numerical results drawn from Theorem 3.4.1 and Theorem 3.4.2. We only focus on the unbalanced multicast tree to study the worst case of RM-cell RTT variations. Since P_j's length is $j+1$ for $j = 1, 2, \cdots, m-1$ (see the unbalanced tree shown in Figure 3.2),

$\tau_u(j, \Delta)$ is the RM-cell RTT for P_j with a length of $j+1$ in an unbalanced tree. Figure 3.4.2 plots P_j's RM-cell RTT $\tau_u(j, \Delta)$ vs. P_j's length $j+1$ and RM-cell interval Δ with tree height $m = 50$ for the two schemes. We observe that for both HBH and SSP schemes RM-cell RTTs $\tau_u(j, \Delta)$'s increase monotonically with path length $j+1$, RM-cell interval Δ, and tree-height m. However, $\tau_u(j, \Delta)$ for the HBH scheme increases much faster, and is always larger, than that for the SSP scheme, and tends to blow up (as high as 1200 τ_h) as $j+1$, Δ, and m increase. In contrast with the HBH scheme, the increase of $\tau_u(j, \Delta)$ for SSP is very limited as $j+1$, Δ, and m get larger. In addition, $\tau_u(j, \Delta)$ for SSP is upper-bounded by $2m = 100 = \tau_{max}$ as shown in Figure 3.4.2, which verifies Theorem 3.4.2. Thus, as shown in Figure 3.4.2, the RM-cell RTT for SSP is virtually independent of path length, RM-cell interval, and multicast-tree height, as compared to the HBH scheme. This is because (1) the synchronization waiting-time is much longer for HBH than that for SSP; (2) the number of forward RM cells required for a feedback RM cell to return from the leaf node to the root in the HBH scheme is proportional to m, while in SSP, any single RM cell can return from the leaf node back to the root by itself.

As analyzed in [6], RM-cell RTTs, or path lengths, have a significant impact on both the bottleneck maximum queue length Q_{max} and the average throughput $\overline{R}$. Due to space limit, we omit the derivations of closed-form expressions for Q_{max} and $\overline{R}$ as functions of RM-cell RTT (which are available on-line in [6]). Instead, we present the numerical solutions of Q_{max} and $\overline{R}$ as the functions of P_j's path length in an unbalanced multicast tree to compare the performance between the HBH and SSP schemes. Assume the multicast-tree bottleneck bandwidth $\mu = 155$ Mbps ≈ 367 cells/ms, $\tau_h = 0.1$ ms, $\Delta = 4\tau_h = 0.4$ ms, and $m = 50$. Figures 3.4.3 and 3.4.3 plot Q_{max} and $\overline{R}$ vs. path length $j+1$ with different rate-gain parameter α [6] for the two different schemes. For HBH, maximum queue length Q_{max} is observed to increase dramatically (see Figure 3.4.3) while the average throughput $\overline{R}$ drops significantly (see Figure 3.4.3) as P_j's path length and tree height m increase. This undesirable trend worsens as α gets larger. In contrast, for SSP with the same parameters

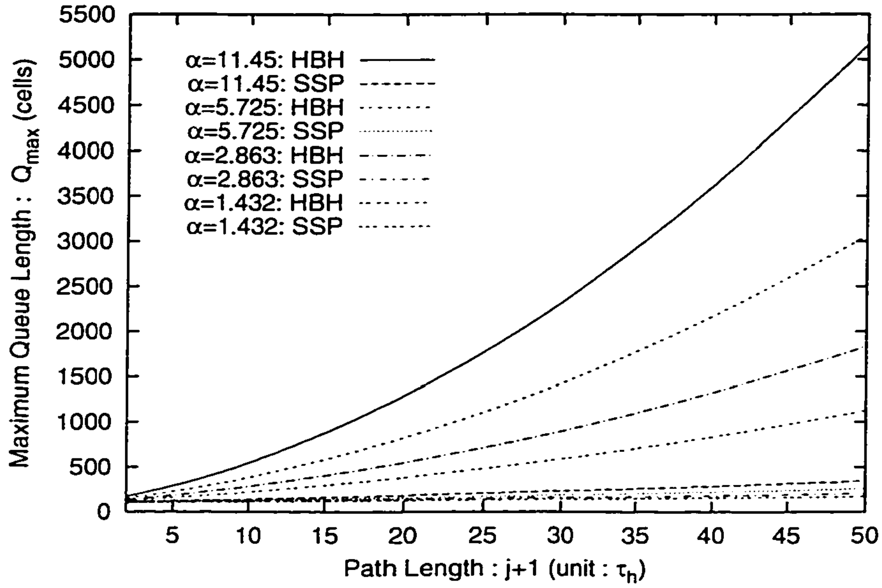


Figure 3.4: Impact of P_j's path length $j+1$, tree height m, $\tau_u(j,\Delta)$ on maximum queue length: Q_{max} vs. $j+1$ ($m=50$).

settings, both Q_{max}'s increase and $\overline{R}$'s drop are very small when $j+1$ and m (even as α varies) increase. Again, Q_{max} and $\overline{R}$ for SSP are found to be virtually independent of the path length and tree height variations. SSP is therefore more scalable than HBH in terms of maximum buffer requirement and average throughput when the multicast-tree topology changes.

3.5 On Selection of RM-Cell Update Interval Δ

Even though the RM-cell RTT for SSP is much smaller than that for HBH, its $\tau(j,\Delta)$ value can be reduced further by properly selecting the RM-cell interval Δ. We now focus on how Δ affects $\tau(j,\Delta)$ and discuss how to select Δ to reduce the SSP's RM-cell RTT.

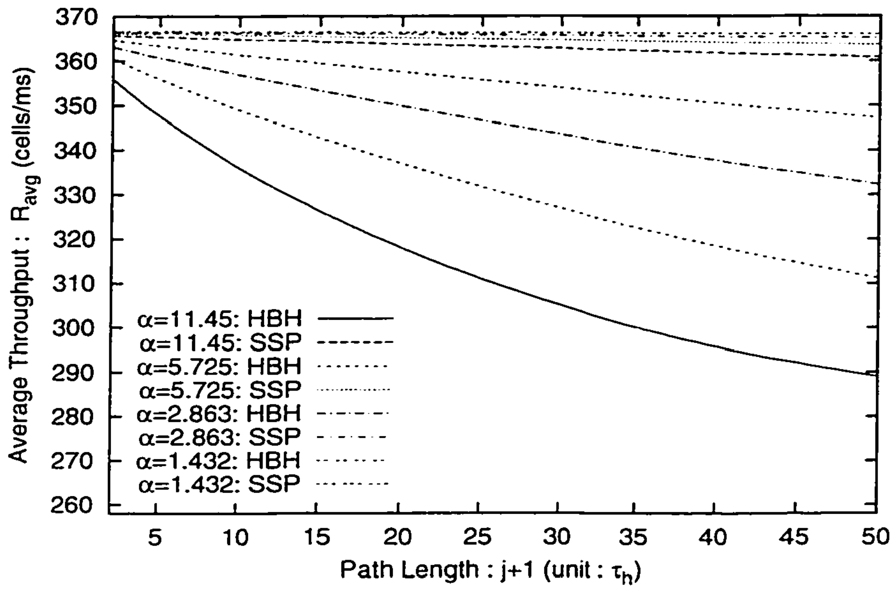


Figure 3.5: Impact of P_j's path length $j+1$, tree height m, and RTT $\tau_u(j, \Delta)$ on the average throughput: $\overline{R}$ vs. $j+1$ with $m = 50$.

3.5.1 Relationships between RM-Cell RTTs and Δ

Unlike unicast, selection of Δ makes a significant impact on all paths' RM-cell RTTs in a multicast tree. To quantify this impact, we introduce the following definitions.

Definition 3.5.1 *If path P_j's feedback RM cell is synchronized only with the feedback RM cells corresponding to the same forward RM cell, then path P_j is said to be **strictly-synchronized**.* ■

P_{m-1} is always strictly-synchronized since it is synchronized only with P_m. The following theorem describes the three *iff* conditions, as a function of Δ, for identifying strictly-synchronized paths.

Theorem 3.5.1 *Let P_j be the j-th path of an unbalanced multicast tree as defined in Lemma 3.4.1* $(1 \leq j \leq m-1)$. *If this multicast tree is flow-controlled by SSP, then the following three claims are equivalent.*

Claim 1: *The number of P_j's RM cells going through the initial state, $k_j^* = 0$, where k_j^* is defined by Eq. (3.5) in Theorem 3.4.2;*

Claim 2: *P_j is strictly-synchronized;*

Claim 3: *P_j's RM-cell RTT attains the maximum: $\tau_u(j, \Delta) = \tau_{max} = 2m$.*

Proof. The proof is provided in Appendix N. ■

Remarks on Theorem 3.5.1: (1) The strictly-synchronized path has the largest RM-cell RTT, and hence, the number of strictly-synchronized paths should be minimized. (2) As shown in Eq. (3.5), a larger Δ results in a larger number of strictly-synchronized paths, and thus the smaller Δ the better.

Definition 3.5.2 *Let W_j be the net waiting time for the P_j's feedback RM cell to synchronize with feedback RM cells via the other paths at all consolidating branch nodes along P_j. If $W_j = 0$, then P_j is said to be* **wait-free synchronized**. ■

Clearly, P_{m-1} is always wait-free synchronized since according to Lemma 3.4.1, a feedback RM cell on a longer path never waits to synchronize with feedback RM cells from shorter paths. Since P_{m-1} is both strictly-synchronized and wait-free synchronized, we exclude P_{m-1} from all the following theorems and treat P_{m-1} separately. The theorem given below provides formulas to determine W_j and establishes the *iff* condition to identify wait-free synchronized paths, all of which are functions of Δ.

Theorem 3.5.2 *Let P_j be the j-th path of an unbalanced multicast tree as defined in Lemma 3.4.1 ($1 \leq j \leq m-2$) and W_j be the net waiting time for the P_j's feedback RM cell to synchronize with feedback RM cells at all consolidating branch nodes along P_j. If this multicast tree is flow-controlled by SSP, then for $1 \leq j \leq m-2$ the following claims hold:*

Claim 1: *P_j's net waiting time W_j for synchronization is upper-bounded by Δ, and W_j is given by:*

$$W_j = 2(m - j - 1) - k_j^* \Delta < \Delta; \tag{3.8}$$

where k_j^* is defined by Eq. (3.5) in Theorem 3.4.2;

<u>Claim 2</u>: *If P_j is strictly-synchronized, then $W_j = 2(m - j - 1) > 0$;*

<u>Claim 3</u>: *P_j is a wait-free synchronized path, i.e., $W_j = 0$ iff $2\,(m - j - 1)$ mod $\Delta = 0$.*

Proof. The proof is provided in Appendix O. ■

Remarks on Theorem 3.5.2: (1) According to Lemma 3.4.2, the wait-free synchronized path has the minimum RM-cell RTT. Thus, the number of wait-free synchronized paths should be maximized. (2) A smaller Δ will lead to a larger number of wait-free synchronized paths. So, a small Δ is desirable.

The following theorem classifies the paths of a multicast tree into three exclusive groups, and provides explicit expressions (as functions of Δ) for calculating the number of paths for each path-group.

Theorem 3.5.3 *Let P_j be the j-th path of an unbalanced multicast tree as defined in Lemma 3.4.1 ($1 \leq j \leq m-2$). If this multicast tree is flow-controlled by SSP, then the entire path set $\mathcal{P} \triangleq \{P_1, P_2, \cdots, P_{m-3}, P_{m-2}\}$ is partitioned into a strictly-synchronized path subset $\mathcal{P}_S$, a wait-free synchronized path subset $\mathcal{P}_N$, and a non strictly-synchronized and non wait-free synchronized path subset $\mathcal{P}_W$, i.e., $\mathcal{P} = \mathcal{P}_S \oplus \mathcal{P}_N \oplus \mathcal{P}_W$, and, furthermore, for $1 \leq \Delta \leq \tau_{max} = 2m$ the following claims hold:*

<u>Claim 1</u>: *The number of strictly-synchronized paths, denoted by S_Δ, is determined by: $S_\Delta \triangleq \|\mathcal{P}_S\| = \lceil \frac{\Delta}{2} \rceil - 1$, where $\|\cdot\|$ denotes the cardinality of a set;*

<u>Claim 2</u>: *The number of wait-free synchronized paths, denoted by N_Δ, is determined by:*

$$N_\Delta \triangleq \|\mathcal{P}_N\| = \begin{cases} \lfloor \frac{2(m-2)}{\Delta} \rfloor, & \text{if } \Delta = even; \\ \lfloor \frac{(m-2)}{\Delta} \rfloor, & \text{if } \Delta = odd; \end{cases} \tag{3.9}$$

<u>Claim 3</u>: *The number of paths which are neither wait-free synchronized nor strictly-synchronized,*

denoted by W_Δ, is determined by:

$$W_\Delta \quad \triangleq \|\mathcal{P}_W\| = \begin{cases} m - \lfloor \frac{2(m-2)}{\Delta} \rfloor - \lceil \frac{\Delta}{2} \rceil - 1, & \textit{if } \Delta = \textit{even;} \\ m - \lfloor \frac{(m-2)}{\Delta} \rfloor - \lceil \frac{\Delta}{2} \rceil - 1, & \textit{if } \Delta = \textit{odd.} \end{cases} \tag{3.10}$$

Proof. The proof is provided in Appendix P. ■

Remarks on Theorem 3.5.3: (1) The number of strictly-synchronized paths is proportional to Δ. (2) The number of wait-free synchronized paths is proportional to $\frac{1}{\Delta}$. (3) If $\Delta = 1$ or 2, then P_j is always wait-free synchronized for all $j = 1, 2, \cdots, m-2$. (4) Taking $\Delta =$ even is preferable in terms of the number of wait-free synchronized paths.

3.5.2 Numerical Evaluation and Discussion

According to Theorem 3.5.3, S_Δ is proportional to Δ while N_Δ is inversely proportional to Δ. Thus, a smaller Δ is desired since strictly-synchronized paths maximize RM-cell RTTs while wait-free synchronization paths minimize RM-cell RTTs. Consider two extreme cases: (1)$\Delta = 1$ (i.e., there is an RM cell traversing per switch-hop) or 2, by Theorem 3.5.3, $S_\Delta = 1$ (P_{m-1} is always strictly-synchronized) and $N_\Delta = m - 1$ (P_{m-1} is always wait-free synchronized), i.e., all paths of interest are wait-free synchronized paths with minimal $\tau_u(j, \Delta) = 2(j+1)$; (2) $\Delta = \tau_{max} = 2m$, by Theorem 3.5.3, $S_\Delta = m - 1$ and $N_\Delta = 1$ (P_{m-1} is always wait-free synchronized). However, the benefits of having larger N_Δ and smaller S_Δ do not come free, the price paid for which is a high bandwidth cost for multicasting RM cells at a higher frequency $\frac{1}{\Delta}$. This introduces a trade-off between $\tau_u(j, \Delta)$ and bandwidth cost for RM cells.

Theorem 3.5.3 suggests that selecting Δ to increase N_Δ is related to tree-height m. As indicated by Eq. (3.9), in order to take advantage of SSP, Δ should not be larger than $m-2$ in which case only P_{m-1} and possibly P_1 (when $\Delta =$ even) are wait-free synchronized paths and more than a half of paths are strictly-synchronized. In Figure 3.6, N_Δ, S_Δ, and W_Δ are plotted against Δ with $m = 50$. We observe that (1) N_Δ decreases as Δ increases; S_Δ is proportional to Δ; W_Δ is not monotonic and reaches its peak value when $N_\Delta = S_\Delta$ and

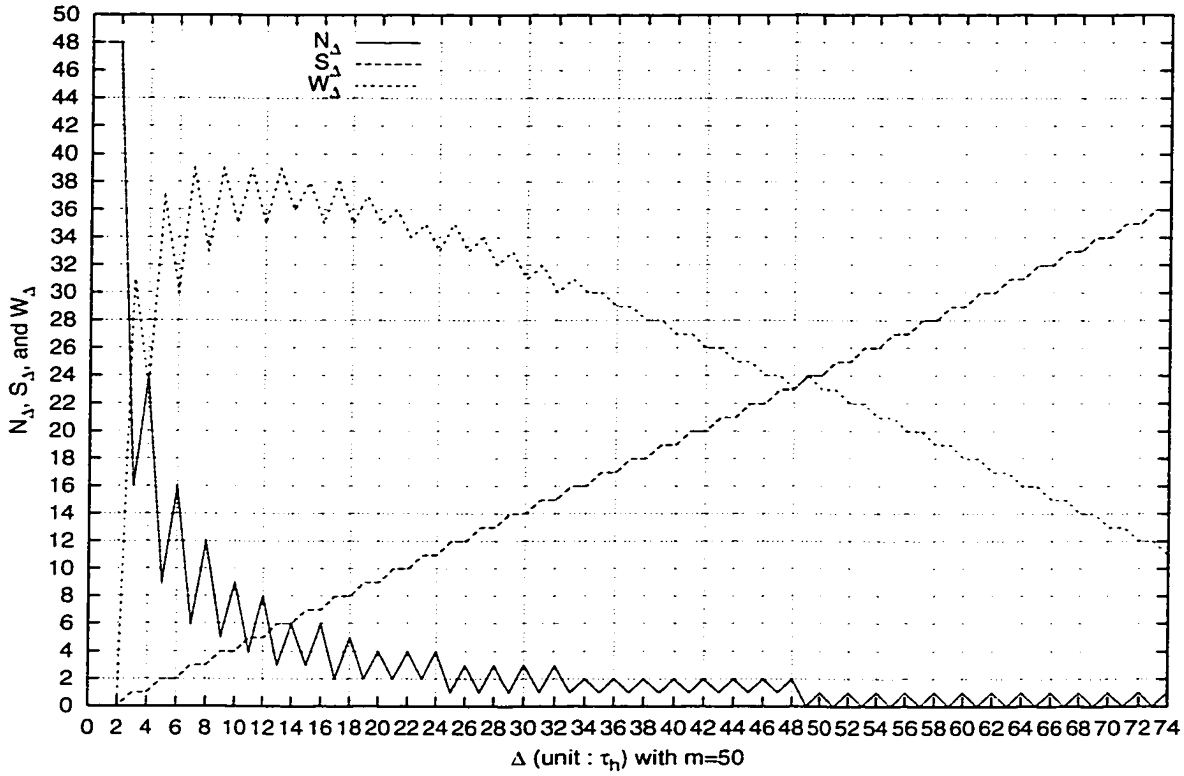


Figure 3.6: N_Δ, S_Δ, and W_Δ vs. Δ ($m = 50$).

$\Delta \in [1, m-2]$. (2) When $\Delta > m-2$, N_Δ becomes very small fluctuating between 0 and 1; and on the other hand, when Δ decreases from $m-2$ to 1, N_Δ increases dramatically. If τ_h is large enough, then taking $\Delta = 2$ will produce the optimal case where all paths become wait-free synchronized. In addition, we also observe that an *even* Δ is preferred since it gives a larger N_Δ than the neighboring values of an odd Δ, which is consistent with Eq. (3.9). Thus, in general, Δ should be taken as an even number within the range of $[2, m-2]$.

Figure 3.7 plots synchronization waiting-time W_j vs. path number j while varying Δ. Although W_j is not a monotonic function of j for a given Δ, W_j increases, on average, as Δ rises. Thus, a smaller Δ is desired to minimize RM-cell RTTs on all paths. We also observe that W_j is a periodic function of j with an amplitude upper-bounded by Δ, verifying Claim 1 of Theorem 3.5.2. Moreover, for a given Δ, there are always some wait-free synchronized

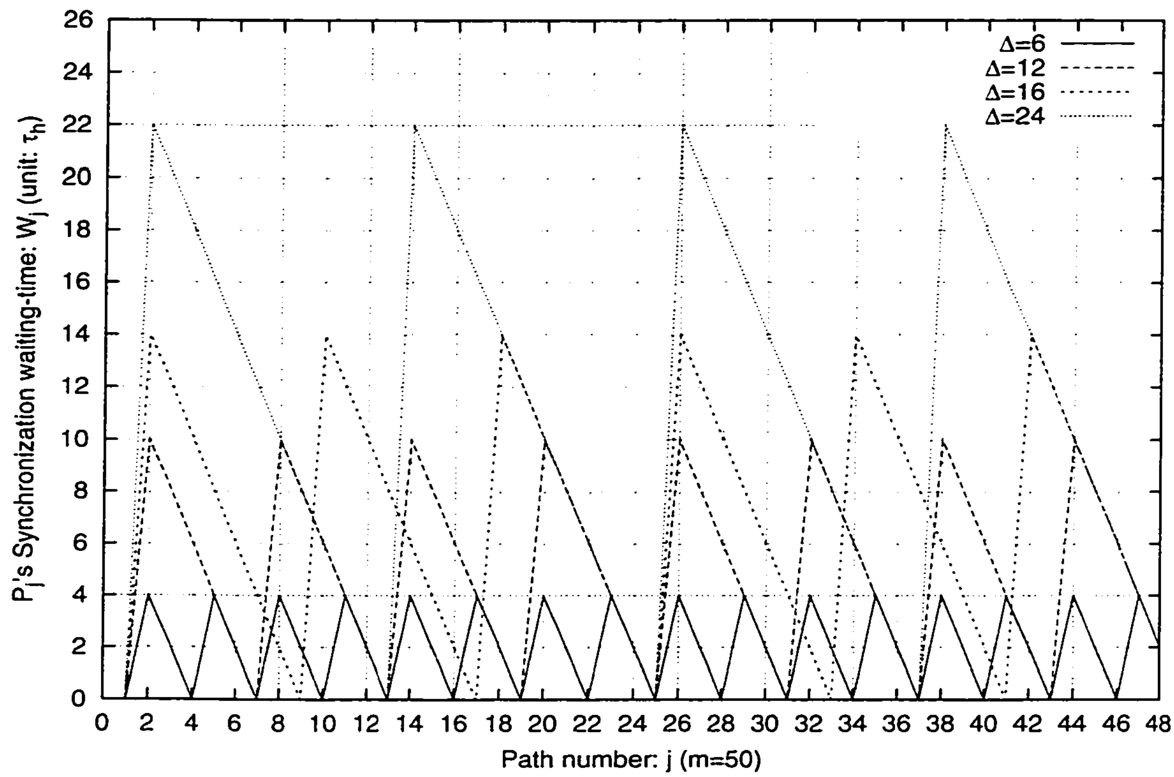


Figure 3.7: W_j vs. path number: j ($m = 50$).

paths ($W_j = 0$). For example, if $\Delta = 6$, there are $N_\Delta = 16$ wait-free synchronized paths, which is consistent with Theorem 3.5.3 and numerical results shown in Figure 3.7 with $m = 50$. Furthermore, Figure 3.7 also shows that a smaller Δ results in a larger number of wait-free synchronized ($W_j = 0$) paths, N_Δ, which also verifies Theorem 3.5.3.

3.6 Conclusion

We developed balanced and unbalanced binary-tree deterministic models to study multicast feedback-synchronization signaling delay characteristics. Applying the proposed binary-tree multicast signaling-delay model, we derived a set of equations to compute each individual path's RTT for a given multicast tree. In contrast with HBH, SSP is shown to be able to efficiently implement multicast signaling and also make the effective RM-cell RTT virtually *independent* of, and hence scalable with, the multicast-tree topology. The numerical analysis has also shown the superiority of SSP over HBH in terms of the multicast signaling delays. We also derived the optimal RM-cell update interval for SSP to minimize RM-cell RTTs for a given multicast tree. While the analysis in this chapter focuses on the feedback-synchronization signaling algorithms for ABR services, it is generic and thus, can be applied to the signaling delay analysis for multicast flow-control algorithms based on *any* feedback-synchronization mechanism.

CHAPTER 4

STATISTICAL DELAY ANALYSIS OF MULTICAST SIGNALING PROTOCOLS

4.1 Introduction

As discussed in Chapter 2, while the delay property of multicast signaling protocols have significant impact on the multicast flow-control performance, little attention has been paid to the delay modeling and analysis for multicast flow control, although it is critically important. To remedy this deficiency, in Chapter 2, we develop the balanced and unbalanced binary-tree models to statically quantify each path's multicast signaling delay. To capture the statistical characteristics of multicast signaling delay when the multicast-tree bottleneck dynamically changes among the multicast-tree paths, in this chapter, we further develop a statistical model to characterize the delay properties for RED- and REM-based multicast flow control, where the random markings at different links are independent.

This chapter is organized as follows. In Section 4.2, we introduce the targeted multicast flow control algorithms and multicasting networks, where the statistical multicast signaling delay modeling and analysis will be applied to. In Section 4.3, we introduce the proposed statistical model and assumptions we made for this proposed statistical model. Section 4.4 derives a set of closed-form expressions to calculate the probability distributions for each path to be the multicast-tree bottleneck. In Section 4.5, we conduct numerical analysis for multicast signaling delays and compare the multicast signaling delays performance between SSP and HBH multicast signaling protocols. Section 4.6 describes the simulation results, verifying the analytical results. In Section 3.5, we derive the optimal RM-cell interval for SSP to minimize the RM-cell RTTs for a given multicast tree. The chapter concludes with Section 4.7.

4.2 The Dynamic Delay Analysis of Multicast-Signaling Protocols in Random-Marking Based Multicast Networks

So far, we only studied the deterministic properties of the feedback-signaling delay for each individual path within a multicast tree under the HBH and SSP schemes. However, in a real-world environment the multicast-tree bottleneck, defined as the most congested path (thus dictating flow-control decisions) of a multicast tree [6], shifts randomly and dynamically from one path to another, depending on the traffic load distribution in the network. To quantitatively analyze the delay characteristics of feedback-synchronization signaling for more realistic multicast scenarios, we now statistically analyze the RM-cell feedback-delay performance of the HBH and SSP schemes across the *entire* multicast tree. The congestion state on each link in the multicast tree are determined by the difference between the aggregate arrival rate and the service rate at that link, or the output-queue length at that link. This congestion state can be specified by the network dynamic congestion status or by flow control algorithms, such as widely cited and used random-marking based schemes: REM [35] or RED [36]. If the targeted multicast flow control algorithms are REM [35] and RED-based or REM and RED-like schemes, then the congestion states or random link-markings at different links are independent [36].

The principle of RED and REM are quite similar, and thus we only give a brief description on RED's the operational procedure that follows below. An RED router operates as follows. It computes the average queue length and when the average queue length exceeds a certain threshold (this threshold can be zero, too), it marks each arrived packet with a certain probability, where the exact probability is a function (e.g., as a linear function used in RED, or an exponential function employed in REM) of the average queue length. The average queue length is calculated using a low-pass filter from instantaneous queue length, which allows the transient bursts in the router. Persistent congestion in the router is reflected by a high average queue length and high marking probability. The resulting high marking probability will signal the traffic source early, and thus detect and control the congestion early. As a result, RED routers (either unicast routers, or multicast routers which will be detailed in Chapter 6) keep the overall throughput high while maintaining a small average queue length, and tolerate transient congestion due to the burstiness caused by window-based flow control scheme TCP. Then the average queue length exceeded a certain threshold, RED routers marks packets at random so that TCP connection or multicast connections back off at different times. This avoid the global synchronization effect of all connections decreasing their sending rates at the same time, resulting in low bandwidth utilization, and maintains high throughput in the routers. Since RED has these excellent features, the IRTF (Internet Research Task Force) has singled out RED as one queue management scheme recommended for rapid deployment throughout the Internet. While the REM and RED schemes are originally proposed for unicasts, they can also be extended to multicast environments as what will shown in Chapter 6. Moreover, unicast and multicast transmissions usually co-exist in a network.

4.3 The Statistical Modeling of Multicast Signaling Delay

4.3.1 The System Model and Assumptions

Our statistical analysis model builds on the recently-proposed Random Early Marking (REM) [35, 37–40] and the widely-cited/used Random Early Detection (RED) schemes [36].[1] The REM and RED schemes can also be extended to multicast environments. Moreover,

[1] The analytical technique developed in this chapter is also applicable to cases where a link's random congestion state is caused by flow control schemes other than REM and RED schemes.

unicast and multicast transmissions usually co-exist in a network. In RED or REM, each router marks the packet's ECN (Explicit Congestion Notification) [41] bit with a probability that is exponential in REM, or proportional in RED, to the average queue length at the output link. To simplify the analysis, our statistical model assumes that the ECN-bit marking operations at all links are independent, an assumption also adopted by REM in [35, 37–40] and RED in [36]. Note that this assumption does not affect the evaluation of the *relative* performance improvement of the SSP scheme over the HBH scheme in terms of the feedback delay.

We will focus only on the RM-cell RTT during the *congestion phase* when the bottleneck paths emerge. Also, the statistical analysis will only concentrate on the unbalanced multicast-tree case because it represents the worst case, and thus its analysis provides a lower bound of performance for the feedback-signaling delay. In contrast, the multicast-tree RM-cell RTT does not change in the balanced-tree case. In addition, our statistical model captures more realistic multicast scenarios by allowing multiple concurrent bottleneck links and paths in a multicast tree. Moreover, to be able to handle any arbitrary size of the unbalanced multicast tree and make the analysis complete, the statistical model allows the multicast-tree height m to be arbitrarily large and include ∞ as its limiting case. To formulate the statistical analysis, we introduce the following definition.

Definition 4.3.1 *The random-marking based unbalanced-multicast binary-tree of height m consists of a set $\mathcal{L}$ of links which satisfy the following conditions:*

C1. *All links in $\mathcal{L}$ are labeled as shown in Figure 4.1(a) for $m < \infty$ (e.g., $m = 4$) and Figure 4.1(b) for $m \to \infty$, respectively, such that*

$$L_i \in \mathcal{L} \triangleq \begin{cases} \{L_1, L_2, \cdots, L_{2m-1}\}, & \text{if } m < \infty; \\ \{L_1, L_2, \cdots, L_\infty\}, & \text{if } m \to \infty; \end{cases} \tag{4.1}$$

C2. $\forall$ *$L_i \in \mathcal{L}$, the probability p_i $(0 < p_i < 1)$ that L_i is marked as a bottleneck link (with*

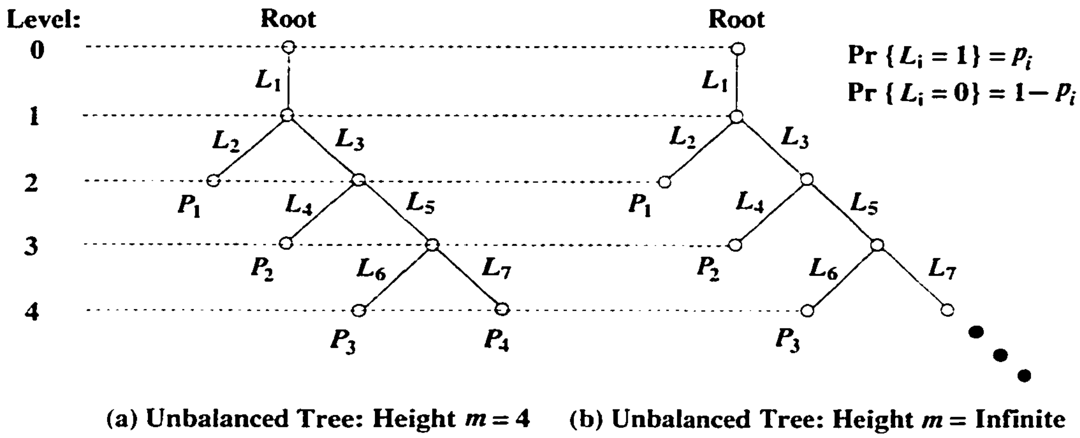


Figure 4.1: Random-marking unbalanced binary-tree model.

the ECN-bit set) is specified by:

$$p_i = \begin{cases} 1 - \phi^{-\gamma \overline{q}_i}, & \textit{if REM is used;} \\ \left[\frac{\overline{q}_i - min_{th}}{max_{th} - min_{th}}\right] p_{max}, & \textit{if RED is used;} \end{cases} \tag{4.2}$$

where $\overline{q}_i$ is average queue size of L_i; $\gamma > 0$ is step size and $\phi > 1$ for REM; p_{max} is the maximum marking probability and max_{th} (min_{th}) is high (low) queue-size thresholds for RED;

C3. *Bottleneck-link (packet) marking events at all links in $\mathcal{L}$ are independent [35–40].* ■

In unicast ABR service, the source rate is regulated by the feedback from the most congested link/switch which has the minimum *available* bandwidth along the path from source to destination. A natural extension of this strategy to multicast ABR service is to adjust the source rate to the available bandwidth share that can be supported by the most congested path, which contains a link/switch or a receiver having the minimum *available* bandwidth across the entire multicast tree. This is the key feature of ABR service, most suitable for data applications, that require *lossless* transmission. Thus, at any given time, the feedback from the most congested path which contains the minimum *available* bandwidth across the multicast tree governs the dynamics of the source rate-control decisions. Consequently, in all the feedback-synchronization signaling algorithms an OR rule (see Figure 3.1) is used at a branch point to consolidate the congestion feedback signals from the different downstream branches. Moreover, since there can be multiple bottleneck links and paths at the same time in the network, we need to identify the path that dictates the dynamics of the entire multicast tree. Clearly, based on the OR rule, the *shortest* bottleneck path in a multicast tree dominates the source's flow-control decisions and the RTT of the flow-control feedback loop. To explicitly model this feature, we introduce the following definition.

Definition 4.3.2 *Among all concurrent bottleneck paths in a multicast tree, the bottleneck path of minimum length is called the* **dominant bottleneck path***, and its RM-cell RTT is called the* **multicast-tree bottleneck RTT**. ■

The multicast-tree bottleneck RTT varies randomly because the location of dominant bottleneck path dynamically drifts as the traffic-load distribution changes with time. Thus, it is important to statistically study the delay properties of feedback-synchronization signaling algorithms.

4.4 Statistical Properties of Feedback Signaling Delays

Based on Definitions 4.3.1 and 4.3.2, the theorem given below derives the probability distribution function of the dominant bottleneck path in a multicast tree.

Theorem 4.4.1 *If an unbalanced multicast binary-tree of height m as defined in Definition 4.3.1 is flow-controlled by the SSP and HBH schemes, respectively, then the following claims hold:*

<u>Claim 1:</u> *If $m \to \infty$, then there exists* ***one and only one*** *dominant bottleneck path, and the probability distribution, denoted by $\psi(P_k, \infty)$, that path P_k becomes the dominant*

bottleneck path, is determined by

$$\psi(P_k,\infty) = (p_{2k-1} + p_{2k} - p_{2k-1}p_{2k}) \prod_{i=1}^{2(k-1)} (1 - p_i) \tag{4.3}$$

where $k = 1, 2, \cdots, \infty$ *and* p_i *is the link* L_i*'s bottleneck-marking probability for* $i = 1, 2, \cdots, \infty$*. Furthermore,* $\psi(P_k, \infty)$ *given in Eq. (4.3) satisfies the following normalization condition:*

$$\lim_{m\to\infty} \sum_{k=1}^{m} \psi(P_k, \infty) = 1. \tag{4.4}$$

<u>**Claim 2:**</u> *If* $m < \infty$*, then there exists at most one dominant bottleneck path, and the probability distribution, denoted by* $\psi(P_k, m)$*, that path* P_k *becomes the dominant bottleneck path, is determined by*

$$\psi(P_k, m) = \begin{cases} p_1 + p_2 - p_1 p_2, & \text{if } k = 1; \\ (p_{2k-1} + p_{2k} - p_{2k-1}p_{2k}) \prod_{i=1}^{2(k-1)} (1 - p_i), & \text{if } k \le m - 1; \\ p_{2m-1} \prod_{i=1}^{2(m-1)} (1 - p_i), & \text{if } k = m; \end{cases} \tag{4.5}$$

where $k = 1, 2, \cdots, m$ *and* p_i *is the link* L_i*'s bottleneck-marking probability for* $i = 1, 2, \cdots, 2m - 1$.

Proof. The proof is provided in Appendix Q. ∎

Remarks on Theorem 4.4.1. By Eq. (4.3), $\lim_{k\to\infty} \psi(P_k, \infty) = 0$, which is expected, since a longer bottleneck path is always dominated by a co-existing shorter bottleneck path. Thus, when $k \to \infty$ as $m \to \infty$, P_∞ is always dominated by a shorter bottleneck path for $0 < p_i < 1$, $i = 1, 2, \cdots, \infty$, where p_i statistically represents the traffic load level at link L_i. That is, $\psi(P_\infty, \infty) = 0$. In addition, by Eq. (4.4) we have $\sum_{k=1}^{\infty} \psi(P_k, \infty) = 1$ which also makes sense because as the unbalanced-tree's height $m \to \infty$ and $0 < p_i < 1$, there always exists (with probability 1) one and only one dominant bottleneck path in a multicast tree. On the other hand, for the case of $m < \infty$, by Eqs. (4.5) and (4.4) we have $\sum_{k=1}^{m} \psi(P_k, m) \leq 1$, implying the possibility that there does not exist any dominant bottleneck path in the multicast tree of height $m < \infty$. This is also expected because the bottleneck link marking probability falls in the range of $0 < p_i < 1$.

Using the probability distributions derived in Theorem 4.4.1, we can obtain the first and second moments of the feedback-synchronization signaling delay for a multicast tree under the HBH and SSP schemes. To simplify the computation, we consider the homogeneous case where

$$p_i = p \qquad \forall i \in \{1, 2, \cdots, 2m-1\}, \tag{4.6}$$

by assuming that the traffic load level, p_i, along all links are statistically at the same level p.[2] This simplification also suffices to assess the relative delay-performance improvement of SSP over the HBH scheme. Under this assumption, the theorem given below derives the probability distribution of the dominant bottleneck path, characterizes its properties, and gives formulas to calculate the first and second order statistics (moments) of feedback-synchronization signaling delay for the SSP and HBH schemes, respectively, for $m < \infty$.

Theorem 4.4.2 *Let an unbalanced multicast binary-tree as defined in Definition 4.3.1 be flow-controlled by the SSP and HBH schemes, respectively, with the RM-cell update interval* Δ. *If the multicast tree has a finite height* $m < \infty$ *and link-marking probability* $0 < p_i = p < 1$, $\forall i \in \{1, 2, \cdots, 2m-1\}$, *then the following claims hold:*

Claim 1: *The probability distribution that the path* P_k *becomes the dominant bottleneck path, denoted by* $\psi(P_k, p, m)$, *is determined by*

$$\psi(P_k, p, m) = \begin{cases} p\,(2-p)\,(1-p)^{2(k-1)}, & \text{if } k \leq m-1; \\ \\ p(1-p)^{2(m-1)}, & \text{if } k = m; \end{cases} \tag{4.7}$$

where $k = 1, 2, \cdots, m$;

[2] The analytical results derived from this special case can be easily extended to a more general case where p_i differs for $i = 1, 2, \cdots, 2m-1$. The derivation procedure for the generalized case remains almost the same as the one derived here.

Claim 2: *For each path P_k, $\psi(P_k, p, m)$ attains the* **unique** *maximum w.r.t. p, which is given by*

$$\psi^*(P_k, p^*, m) = \begin{cases} \frac{1}{k}\left(1-\frac{1}{k}\right)^{k-1}, & \text{if } k \leq m-1; \\ \frac{1}{2m-1}\left(\frac{2m-2}{2m-1}\right)^{2(m-1)}, & \text{if } k = m; \end{cases} \tag{4.8}$$

where the **unique** *bottleneck-link marking probability maximizer, p^*, is determined by*

$$p^* \triangleq \arg \max_{0<p<1} \psi(P_k, p, m) = \begin{cases} 1-\sqrt{\frac{k-1}{k}}, & \text{if } k \leq m-1; \\ \frac{1}{2m-1}, & \text{if } k = m; \end{cases} \tag{4.9}$$

where $k = 1, 2, \cdots, m$;

Claim 3: *The means of multicast-tree RM-cell RTT, denoted by $\overline{\tau}_{SSP}(m)$ and $\overline{\tau}_{HBH}(m)$ for the SSP and HBH schemes, respectively, are determined by:*

$$\overline{\tau}_{SSP}(m) = 2m\left[1-(1-p)^{2(m-1)}\right] - \Delta\left(2p-p^2\right)\sum_{k=1}^{m-1}\left\lfloor\frac{2(m-k-1)}{\Delta}\right\rfloor(1-p)^{2(k-1)}$$
$$+2mp(1-p)^{2(m-1)}; \tag{4.10}$$

$$\overline{\tau}_{HBH}(m) = 2\left[1-(1-p)^{2(m-1)}\right] + \frac{\Theta(\Delta)}{2p-p^2}\left[(m-1)(1-p)^{2m} - m(1-p)^{2(m-1)} + 1\right]$$
$$+p(1-p)^{2(m-1)}\left[2+(m-1)\Theta(\Delta)\right]; \tag{4.11}$$

where $\Theta(\Delta)$ is defined by Eq. (3.2);

Claim 4: *The variances of multicast-tree RM-cell RTT, denoted by $\sigma^2_{SSP}(m)$ and $\sigma^2_{HBH}(m)$*

for the SSP and HBH schemes, respectively, are determined by:

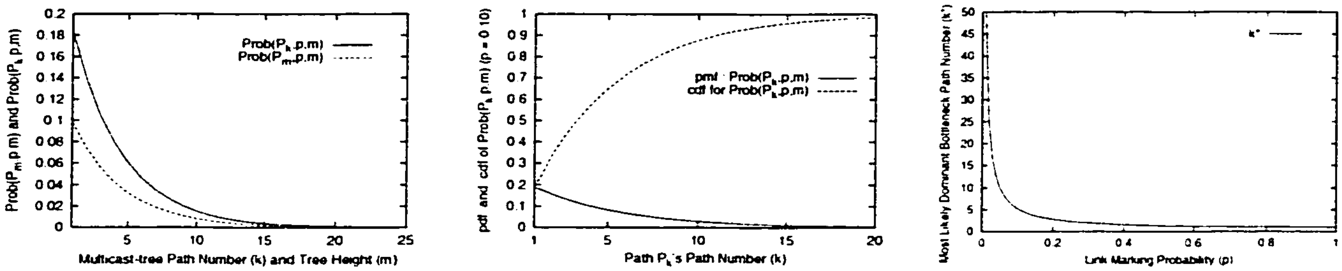
$$\begin{aligned}
\sigma^2_{SSP}(m) &= 4m^2\left[1-(1-p)^{2(m-1)}\right] - (2p-p^2)\left\{4m\Delta\sum_{k=1}^{m-1}\left\lfloor\frac{2(m-k-1)}{\Delta}\right\rfloor\right. \\
&\quad \left.\cdot(1-p)^{2(k-1)} - \Delta^2\sum_{k=1}^{m-1}\left\lfloor\frac{2(m-k-1)}{\Delta}\right\rfloor^2(1-p)^{2(k-1)}\right\} \\
&\quad -\left\{2m\left[1-(1-p)^{2(m-1)}\right] - \Delta\left(2p-p^2\right)\sum_{k=1}^{m-1}\left\lfloor\frac{2(m-k-1)}{\Delta}\right\rfloor\right. \\
&\quad \left.\cdot(1-p)^{2(k-1)} + 2mp(1-p)^{2(m-1)}\right\}^2 + 4m^2p(1-p)^{2(m-1)}.
\end{aligned} \tag{4.12}$$

$$\begin{aligned}
\sigma^2_{HBH}(m) &= 4\left[1-(1-p)^{2(m-1)}\right] + \frac{4\Theta(\Delta)}{2p-p^2}\left[1-m(1-p)^{2(m-1)}+(m-1)\right. \\
&\quad \left.\cdot(1-p)^{2m}\right] + \frac{\Theta^2(\Delta)}{(2p-p^2)^2}\left\{(2p-p^2)\left[1-m^2(1-p)^{2(m-1)}\right.\right. \\
&\quad \left.+(m^2-1)(1-p)^{2m}\right] + 2\left[(1-p)^2-m(1-p)^{2m}+(m-1)\right. \\
&\quad \left.\left.\cdot(1-p)^{2(m+1)}\right]\right\} + p(1-p)^{2(m-1)}\left[2+(m-1)\Theta(\Delta)\right]^2 \\
&\quad -\left\{\frac{\Theta(\Delta)}{2p-p^2}\left[(m-1)(1-p)^{2m}-m(1-p)^{2(m-1)}+1\right]\right. \\
&\quad +2\left[1-(1-p)^{2(m-1)}\right] + (1-p)^{2(m-1)}p \\
&\quad \left.\cdot\left[2+(m-1)\Theta(\Delta)\right]\right\}^2,
\end{aligned} \tag{4.13}$$

where $\Theta(\Delta)$ *is defined by Eq. (3.2).*

Proof. The proof is provided in Appendix R. ■

Remarks on Theorem 4.4.2: Under the assumption that each link's bottleneck marking probability $p = p_i$, $\forall i \in \{1, 2, \cdots, 2m-1\}$, and observing Eq. (4.7), $\psi(P_k, p, m)$ is found to be a strictly monotonic decreasing function of path length k and the multicast-tree height m. Figure 4.2(a) plots the *probability mass function (pmf)* $\psi(P_k, p, m)$ against k and m, also confirming the above observation. This is not surprising because a longer bottleneck path is more likely to be dominated by a shorter one. In fact, this is a desired feature for



(a) $\psi(P_k, p, m)$ vs. k and m (b) *pmf* $\psi(P_k, p, m)$ & its *cdf* vs. k. (c) k^* vs. p.

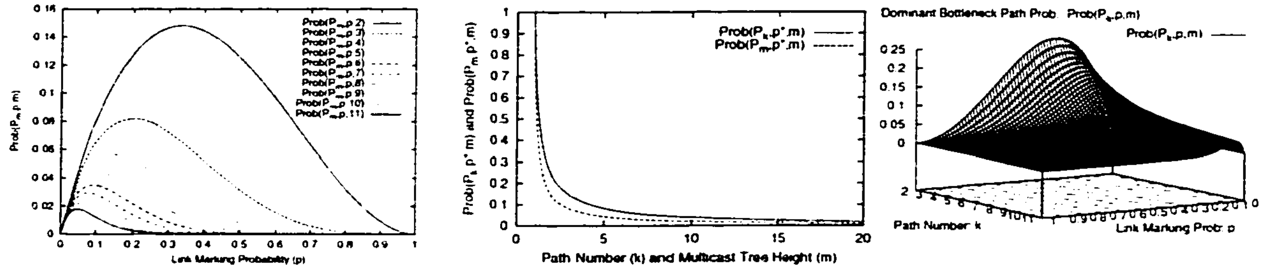
Figure 4.2: Probability distributions of dominant bottleneck path.

the SSP scheme because the SSP's effective multicast-tree RM-cell RTT is upper-bounded by the maximum RM-cell RTT and is virtually independent of the multicast-tree height. Figure 4.2(b) also plots the *cumulative distribution function (cdf)* for $\psi(P_k, p, m)$, which converges to 1 as $k, m \to \infty$, confirming that $\psi(P_k, p, m)$ is a valid *pmf*.

Eq. (4.7) also indicates that for a given bottleneck path P_k, its dominant bottleneck path probability $\psi(P_k, p, m)$ is not a monotonic function of link-marking probability p. $\psi(P_k, p, m)$ attains the maximum value, $\psi^*(P_k, p^*, m)$ determined by Eq. (4.8), as a function of p, which statistically reflects the network traffic load. Solving the first part of Eq. (4.9) for k with a given p, we obtain

$$k^* \triangleq \frac{1}{p(2-p)}, \tag{4.14}$$

where P_{k^*} is "most likely" to be the dominant bottleneck path for the given link marking probability p. When p departs from p^*, $\psi(P_k, p, m)$ converges to zero as either $p \to 0$ or $p \to 1$, as shown in Eq. (4.7). This is expected because a small p implies that the entire network traffic load is low, driving the most likely dominant bottleneck path P_{k^*} towards the longest path, while a large p means that the entire network traffic load is heavy, making the most likely dominant bottleneck path P_{k^*} shift towards the shortest path. If the given path has a length somewhere between the longest and shortest paths, the probability for the path to be a dominant bottleneck path converges to zero as $p \to 0$ or 1. On the other hand, p^* and $\psi^*(P_k, p^*, m)$ are both the monotonic decreasing functions of path length k,



(a) $\psi(P_m, p, m)$ vs. p (b) $\psi^*(P_k, p^*, m)$ vs. p (c) $\psi(P_k, p, m)$ vs. (p, k)

Figure 4.3: Properties of dominant bottleneck path probability-distribution functions.

because in general $\psi(P_k, p, m)$ is a strictly decreasing function of k and, when k increases, the link-marking probability p must decrease to ensure a longer path to be the most likely dominant bottleneck path, which, in turn, reduces p^*. Figure 4.2(c) plots the path number k^* of the "most-likely" dominant bottleneck path against p based on Eq. (4.14), and shows that k^* decreases as p increases. That is, the higher the network traffic load, the shorter the most likely dominant bottleneck path. This makes SSP multicast signaling scheme based on REM or RED very suitable for multicast flow control since multicast-tree RM-cell RTT statistically adapts to network traffic load variations. Figure 4.3(b) plots $\psi^*(P_k, p^*, m)$ against path length k and multicast-tree height m. We observe that $\psi^*(P_k, p^*, m)$ drops very quickly when the path length k and multicast tree height m rise, which makes the longer path to have a relatively smaller probability to become the most likely dominant bottleneck path as compared to the shorter path (also see Figure 4.3(c)).

Figure 4.3(a) demonstrates how the network traffic load and multicast-tree height affect the dominant bottleneck path probability by plotting $\psi(P_k, p, m)$ vs. p with different m values. Figure 4.3(a) clearly shows that there exists a unique maximum $\psi^*(P_k, p^*, m)$ for any given m. As m increases, $\psi^*(P_k, p^*, m)$ decreases, confirming the above observations. In Figure 4.3(c), $\psi(P_k, p, m)$ is plotted as a function of two independent variables, p and k. We observe that for each given path P_k, $\psi(P_k, p, m)$ attains its unique maximum at p^* while both $\psi^*(P_k, p^*, m)$ and p^* are monotonic decreasing functions of k. This also confirms our analytical findings.

$\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$ are important performance metrics for the feedback-synchronization signaling since they represent the average RM-cell RTT of a multicast tree. Clearly, small $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$ are desired because a small feedback delay can improve feedback accuracy and system responsiveness. Eqs. (4.10) and (4.11) indicate that both $\overline{\tau}_{SSP}(m)$ and $\overline{\tau}_{HBH}(m)$ are functions of Δ and m. So, the selection of Δ will affect the multicast tree's average RTT. On the other hand, $\sigma^2_{HBH}(m)$ and $\sigma^2_{SSP}(m)$ represent the variation amplitudes around the average tree RM-cell RTT for HBH and SSP schemes, respectively. Also, small $\sigma^2_{HBH}(m)$ and $\sigma^2_{SSP}(m)$ are desired, because they impact the stability and transient performance of flow control. Likewise, Eqs. (4.12) and (4.13) indicate that both $\sigma^2_{SSP}(m)$ and $\sigma^2_{HBH}(m)$ are functions of Δ and m. So, the selection of Δ will also affect the variation of multicast-tree's average RM-cell RTT.

The next corollary that follows directly from Theorem 4.4.2 for the homogeneous case of $p_i = p$, $\forall i$, by letting $m \to \infty$, derives the probability distribution of the dominant bottleneck path, analyzes its properties, and provides expressions for the statistical properties of feedback delay for HBH and SSP, respectively, when $m \to \infty$.

Corollary 4.4.1 *Let an unbalanced multicast binary-tree as defined in Definition 4.3.1 be flow-controlled by the SSP and HBH schemes, respectively, with the RM-cell update interval* Δ. *If the multicast-tree's height* $m \to \infty$ *and link-marking probability* $0 < p_i = p < 1$, $\forall i \in \{1, 2, \cdots, \infty\}$, *then the following claims hold:*

Claim 1: *The probability distribution that P_k becomes the dominant bottleneck path, denoted by $\psi(P_k, p, \infty)$, is determined by*

$$\psi(P_k, p, \infty) = p(2-p)(1-p)^{(2k-1)} \tag{4.15}$$

where $k = 1, 2, \cdots, \infty$. Also, $\psi(P_k, p, \infty)$ given in Eq. (4.15) satisfies the following normalization condition:

$$\lim_{m\to\infty} \sum_{k=1}^{m} \psi(P_k, p, \infty) = 1; \tag{4.16}$$

Claim 2: *For each path P_k, $\psi(P_k, p, \infty)$ attains the unique maximum given by*

$$\psi^*(P_k, p^*, \infty) = \begin{cases} \frac{1}{k}\left(1-\frac{1}{k}\right)^{k-1}, & \text{if } k < \infty; \\ 0, & \text{if } k \to \infty; \end{cases} \tag{4.17}$$

where the bottleneck-link marking probability maximizer p^* is determined by

$$p^* \triangleq \arg \max_{0<p<1} \psi(P_k, p, \infty) = \begin{cases} 1 - \sqrt{\frac{k-1}{k}}, & \text{if } k < \infty; \\ 0, & \text{if } k \to \infty; \end{cases} \tag{4.18}$$

where $k = 1, 2, \cdots, \infty$;

Claim 3: *The means of multicast-tree RM-cell RTT, denoted by $\overline{\tau}_{HBH}(\infty)$ and $\overline{\tau}_{SSP}(\infty)$ for the HBH and SSP schemes, respectively,* exist *and are determined by:*

$$\overline{\tau}_{HBH}(\infty) = \frac{4p - 2p^2 + \Theta(\Delta)}{2p - p^2}; \tag{4.19}$$

$$\overline{\tau}_{SSP}(\infty) = \lim_{m\to\infty} \left\{ 2m - \frac{\Delta\,(2p-p^2)}{(1-p)^2} \sum_{k=1}^{m-1} \left\lfloor \frac{2(m-k-1)}{\Delta} \right\rfloor (1-p)^{2k} \right\}; \tag{4.20}$$

where $\Theta(\Delta)$ is defined by Eq. (3.2);

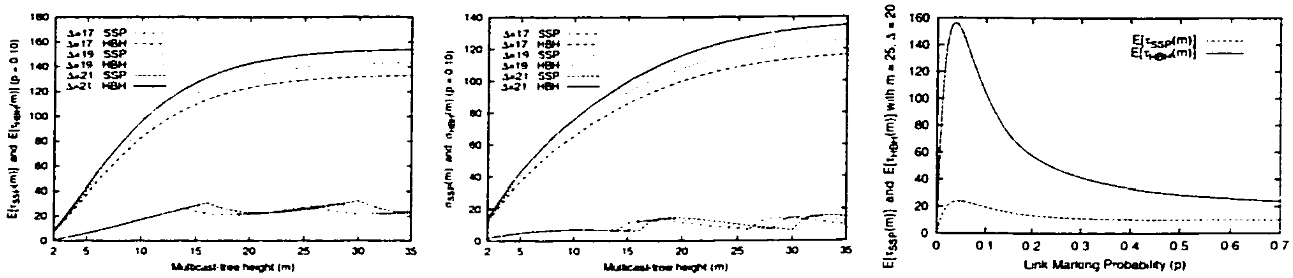
Claim 4: *The variances of multicast-tree RM-cell RTT, denoted by $\sigma^2_{HBH}(\infty)$ and $\sigma^2_{SSP}(\infty)$ for the HBH and SSP schemes, respectively,* exist *and are determined by:*

$$\sigma^2_{HBH}(\infty) = \frac{(1-p)^2\Theta^2(\Delta)}{(2-p)^2p^2}, \tag{4.21}$$

$$\begin{aligned} \sigma^2_{SSP}(\infty) = \lim_{m\to\infty} \Bigg\{ & 4m^2 - \frac{(2p-p^2)}{(1-p)^2} \Bigg\{ 4m\Delta \sum_{k=1}^{m-1} \left\lfloor \frac{2(m-k-1)}{\Delta} \right\rfloor (1-p)^{2k} - \Delta^2 \\ & \cdot \sum_{k=1}^{m-1} \left\lfloor \frac{2(m-k-1)}{\Delta} \right\rfloor^2 (1-p)^{2k} \Bigg\} - \Bigg\{ 2m - \frac{\Delta\,(2p-p^2)}{(1-p)^2} \\ & \cdot \sum_{k=1}^{m-1} \left\lfloor \frac{2(m-k-1)}{\Delta} \right\rfloor (1-p)^{2k} \Bigg\}^2 \Bigg\}. \end{aligned} \tag{4.22}$$

where $\Theta(\Delta)$ is defined by Eq. (3.2);

Proof. The proof is provided in Appendix S. ■



(a) $\overline{\tau}_{SSP}(m)$, $\overline{\tau}_{HBH}(m)$ vs. m (b) $\sigma_{SSP}(m)$, $\sigma_{HBH}(m)$ vs. m (c) $\overline{\tau}_{SSP}(m)$, $\overline{\tau}_{HBH}(m)$ vs. p

Figure 4.4: Comparison: means and variances of multicast-tree RTT between HBH and SSP schemes

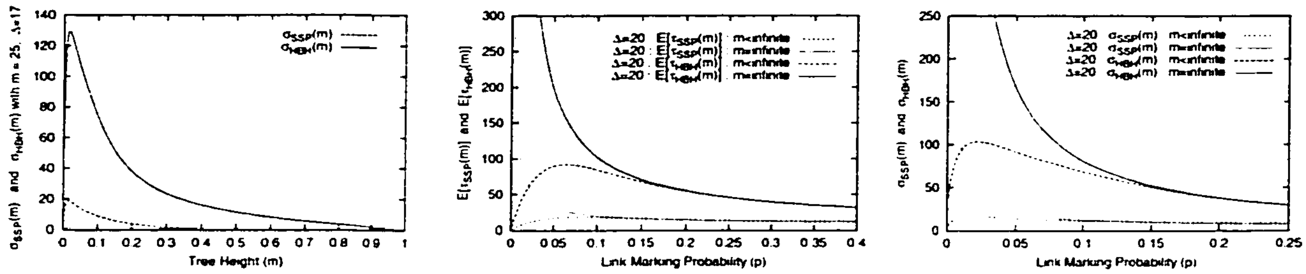
Remarks on Corollary 4.4.1: While m does not attain ∞ in real networks, Corollary 4.4.1 ensures the existence and convergence of *finite* means and variances for the dominant bottleneck-path probability distribution derived in Theorem 4.4.1. This makes our statistical analysis complete and meaningful. This corollary also states the trend of means and variances of the SSP and HBH schemes when m is large. For instance, from Eq. (4.19) we observe that $\overline{\tau}_{HBH}(\infty)$ is proportional to the RM-cell interval Δ (or $\Theta(\Delta)$) for a given p. In contrast, from Eq. (4.20), we observe that $\overline{\tau}_{SSP}(\infty)$ is upper-bounded by the maximum RM-cell RTT, $2m$, regardless of p. Likewise, Eq. (4.21) indicates that $\sigma^2_{HBH}(\infty)$ is proportional to $\Theta^2(\Delta)$ or Δ^2 while $\sigma^2_{SSP}(\infty)$ is also upper-bounded.

4.5 Numerical Comparison of Statistical Properties for SSP and HBH

Using the analytical results derived in Section 4.4 we numerically compare the statistical delay properties for HBH and SSP. Figure 4.4(a) plots $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$, respectively, against the multicast-tree height m for different RM-cell interval Δ's. From Figure 4.4(a) we observe that $\overline{\tau}_{HBH}(m)$ is much larger ($\approx$ 6 times), and increases much faster, than $\overline{\tau}_{SSP}(m)$. Moreover, $\overline{\tau}_{HBH}(m)$ is more sensitive to Δ than $\overline{\tau}_{SSP}(m)$. Figure 4.4(a) also shows that $\overline{\tau}_{SSP}(m)$ — unlike RTT $\overline{\tau}_{HBH}(m)$ — is virtually independent of m. Figure 4.4(b) plots $\sigma_{HBH}(m)$ and $\sigma_{SSP}(m)$ against m while varying Δ. From Figure 4.4(b), $\sigma_{HBH}(m)$ is found to be much larger ($\approx$ 6 times), and increase much faster, than $\sigma_{SSP}(m)$ as m increases. Again, $\sigma_{HBH}(m)$ is much more sensitive to Δ than $\sigma_{SSP}(m)$. Thus, the multicast-tree RM-cell RTT for SSP scales much better than that for HBH with respect to the multicast-tree height and structure. Also, as illustrated in Figure 4.4(b), SSP's multicast-tree RTT variation $\sigma_{SSP}(m)$ is virtually independent of m as compared to HBH's multicast-tree RTT variation, $\sigma_{HBH}(m)$.

Setting tree height $m = 25$ and RM-cell RTT interval $\Delta = 20$, Figure 4.4(c) plots $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$ against link marking probability p, which statistically represents the network traffic load. We observe that both $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$ have their respective unique maximum, which is expected because $\psi(P_k, p, m)$ has the unique maximum with respect to p. Moreover, the maximum of $\overline{\tau}_{HBH}(m)$ is found to be about 8 times larger than that of $\overline{\tau}_{SSP}(m)$ while the maximizer ($p = 0.037$) of $\overline{\tau}_{HBH}(m)$ is slightly smaller than that ($p = 0.044$) of $\overline{\tau}_{SSP}(m)$. So, the SSP significantly outperforms the HBH in terms of the first moment of multicast-signaling delays. When p increases beyond the maximizer, both $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$ decrease quickly since the dominant bottleneck path tends to be short as the network traffic load increases. This observation states the fact that the multicast-tree RM-cell RTT of the SSP multicast signaling scheme based on REM or RED can statistically adapt itself to the network traffic-load variation dynamically. Figure 4.5(a) plots $\sigma_{HBH}(m)$ and $\sigma_{SSP}(m)$ against p with $m = 25$ and $\Delta = 17$. Likewise, both $\sigma_{HBH}(m)$ and $\sigma_{SSP}(m)$ are observed to have their own unique maximum, which is also due to the uniqueness of maximum for $\psi(P_k, p, m)$ over p. Also, $\sigma_{HBH}(m)$ is found to be much larger (about 7 times) than $\sigma_{SSP}(m)$ while the maximizer ($p = 0.017$) of $\sigma_{HBH}(m)$ is slightly larger than that ($p = 0.015$) of $\sigma_{SSP}(m)$. This observation indicates that the multicast-tree RM-cell RTT for SSP is statistically much stabler than that for HBH in terms of multicast signaling delay variations. Moreover, the dynamics of $\sigma_{HBH}(m)$ and $\sigma_{SSP}(m)$ with traffic load p varying behave in a manner similar to those of $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$.

To examine the properties of the first and second moments of the multicast-tree bottle-



(a) $\sigma_{SSP}(m)$, $\sigma_{HBH}(m)$ vs. p (b) $\overline{\tau}_{SSP}(\infty)$, $\overline{\tau}_{HBH}(\infty)$ vs. p (c) $\sigma_{SSP}(\infty)$, $\sigma_{HBH}(\infty)$ vs. p

Figure 4.5: Statistical and asymptotic properties of multicast-tree RTT for HBH and SSP as $m \to \infty$.

neck path RM-cell RTT when m is large, their plots as the function of p (with $\Delta = 20$, and $m = 25$ if $m < \infty$) in Figures 4.5(b) and (c) show the trends of $\overline{\tau}_{HBH}(m) \to \overline{\tau}_{HBH}(\infty)$, $\overline{\tau}_{SSP}(m) \to \overline{\tau}_{SSP}(\infty)$, $\sigma_{HBH}(m) \to \sigma_{HBH}(\infty)$, and $\sigma_{SSP}(m) \to \sigma_{SSP}(\infty)$, asymptotically as $m \to \infty$. We made the following observations.

O1. As $m \to \infty$, the extreme points for $\overline{\tau}_{HBH}(\infty)$, $\overline{\tau}_{SSP}(\infty)$, $\sigma_{HBH}(\infty)$, and $\sigma_{SSP}(\infty)$ disappear, and become monotonic decreasing functions of p. This is expected since $p^* = 0$ and $\psi^*(P_k, p^*, \infty) = 0$ as $k, m \to \infty$ as shown in Eqs. (4.18) and (4.17).

O2. $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$ converge to $\overline{\tau}_{HBH}(\infty)$ and $\overline{\tau}_{SSP}(\infty)$, respectively, and both are lower-bounded by the same bound for each scheme, as $p \to 1$. This also confirms our analytical results, because $\lim_{p\to 1} \overline{\tau}_{HBH}(m) = \lim_{p\to 1} \overline{\tau}_{HBH}(\infty) = 2 + \Theta(\Delta)$, which is the lower bound of HBH's RM-cell RTT given by Eq. (3.1) for path P_1, and $\lim_{p\to 1} \overline{\tau}_{SSP}(m) = \lim_{p\to 1} \overline{\tau}_{SSP}(\infty) = 2m - \Delta \left\lfloor \frac{2(m-2)}{\Delta} \right\rfloor$, which is the lower bound of SSP's RM-cell RTT given by Eq. (3.6) for path P_1.

O3. $\sigma_{HBH}(m)$ and $\sigma_{SSP}(m)$ converge to $\sigma_{HBH}(\infty)$ and $\sigma_{SSP}(\infty)$ as $m \to \infty$, respectively, and both converge to 0, as $p \to 1$. This also confirms our analytical results, because from Eqs. (4.13) and (4.21), we have $\lim_{p\to 1} \sigma^2_{HBH}(m) = \lim_{p\to 1} \sigma^2_{HBH}(\infty) = 0$, and from Eqs. (4.12) and (4.22), we have $\lim_{p\to 1} \sigma^2_{SSP}(m) = \lim_{p\to 1} \sigma^2_{SSP}(\infty) = 0$.

O4. When m is large, $\overline{\tau}_{HBH}(m)$ and $\overline{\tau}_{SSP}(m)$ drop very quickly as p increases. For instance, when $p \geq 0.1$, i.e., when network is busy for more than 10% of the time,

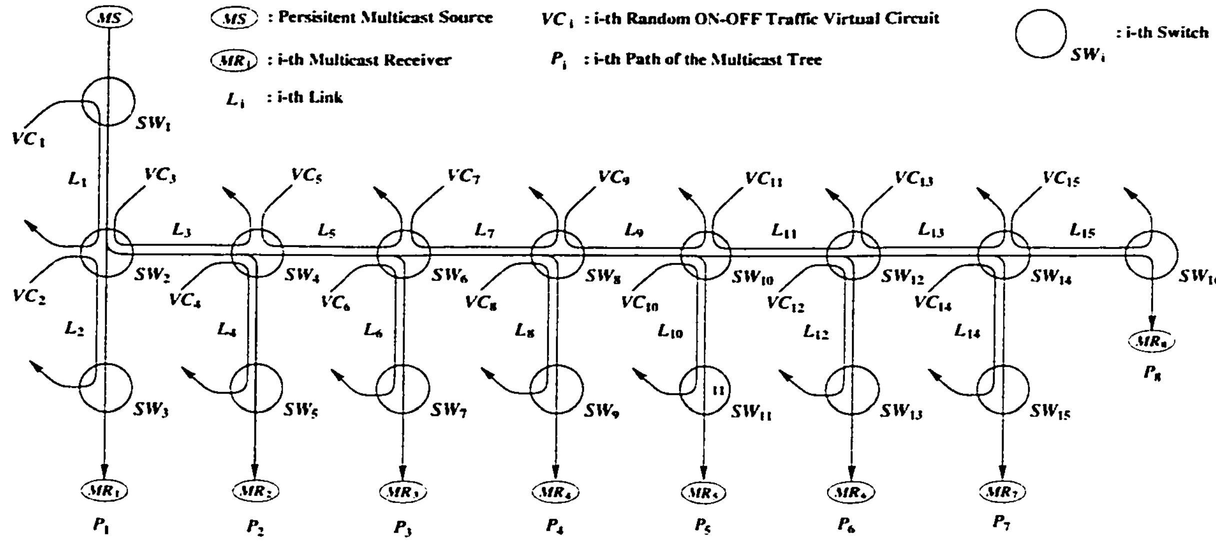


Figure 4.6: Simulation model for delay analysis of unbalanced-tree bottleneck RTT with $m = 8$.

$\overline{\tau}_{SSP}(m)$ already converges closely to the lower bound ($\lim_{p\to 1}\overline{\tau}_{SSP}(m) = 2m - \Delta\left\lfloor\frac{2(m-2)}{\Delta}\right\rfloor = 10$) for the case of $m = 25$ and $\Delta = 20$. In contrast, $\overline{\tau}_{HBH}(m)$ does not converge closely to its lower bound ($\lim_{p\to 1}\overline{\tau}_{HBH}(m) = 2 + \Delta = 22$) until the network traffic load is beyond 40%–50% for the same case. Likewise, a similar behavior is found to hold for $\sigma_{HBH}(m)$ and $\sigma_{SSP}(m)$. This reveals that the SSP's multicast-tree RM-cell RTT and its variation converge to the lower bound much faster than the HBH's. Thus, the SSP scheme adapts to network traffic-load much faster than the HBH scheme in terms of multicast-tree RM-cell RTT.

4.6 Simulation Results

We simulate the network with concurrent multiple multicast/unicast VCs (Virtual Circuits) and multiple bottlenecks to study the statistical behavior of SSP signaling delay and compare it with HBH.

This simulation study focuses on the unbalanced multicast-tree case because it represents the worst, but general, case for signaling delay variations, and thus, provides a lower bound of the delay performance. In contrast, the balanced-tree case is trivial and can be treated as a special case of the unbalanced-tree case. The simulated network in Figure 4.6 consists of 16 switches, $SW_1, SW_2, \cdots, SW_{16}$, connected via 15 links $L_1, L_2, \cdots, L_{15}$ as an unbalanced multicast tree of height $m = 8$. As shown in Figure 4.6, the network contains one multicast connection with a persistent ABR traffic source, starting from the sender MS to 8 receivers $MR_1, MR_2, \cdots, MR_8$ through the multicast tree and forming 8 paths $P_1, P_2, \cdots, P_8$. We also set up 15 independent *random* ON-OFF unicast VCs, each of which is represented by VC_i corresponding to link L_i for $i = 1, 2, \cdots, 15$. VC_i functions as independent random cross-traffic sharing L_i's bandwidth with the multicast connection. The activity intensity of the cross-traffic generated by VC_i determines the congestion marking probability p_i for link L_i. When these cross-traffic sessions randomly switch between ON and OFF states, the multicast-tree bottleneck changes randomly from one path to another.

We implemented the simulation model by using the NetSim event-driven simulator [42] where we set all links to have identical bandwidth $\mu_i = \mu = 155$ Mbps and link (or hop) delay $\tau_h = 1$ ms (millisecond). Thus, all paths' RTTs, $\tau_u^{HBH}(j, \Delta)$ and $\tau_u^{SSP}(j, \Delta)$, are given by Eqs. (3.1) and (3.6) for HBH and SSP (steady-state), respectively, which are obtained and listed in Table 4.1. The last row in Table 4.1 gives the physically limiting minimum for the RTT, $\tau_{min}^{Limit}(j)$, of each path of the simulated multicast tree. The RM-cell interval is set to 6 ms.

Path Name	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
$\tau_u^{HBH}(j, \Delta)$	8	14	20	26	32	38	44	44
$\tau_u^{SSP}(j, \Delta)$	4	10	10	10	16	16	16	16
$\tau_{min}^{Limit}(j)$	4	6	8	10	12	14	16	16

Table 4.1: RTT (unit: ms) for each path for the simulated network model.

The link congestion marking probabilities, p_i ($i = 1, \cdots, 15$), are generated by 15 independent $[0, 1]$-uniform random-number generators, R_i, which control the 15 cross-traffic VC_i's ON-OFF states and their activity intensities. The entire simulation observation time T is divided into N repeated observation slots T_k, $k = 1, 2, \cdots, N$, and $T = \sum_{k=1}^{N} T_k$. At the beginning of each observation slot T_k, $k = 1, 2, \cdots, N$, VC_i, $i = 1, 2, \cdots, 15$, enters an ON (OFF) state and stay there for a period of T_k if $R_i \leq p_i$ ($R_i > p_i$), such that

$$\Pr\{VC_i = \text{ON}\} = \Pr\{R_i \leq p_i\} = \int_0^{p_i} 1\, du = p_i \tag{4.23}$$

and

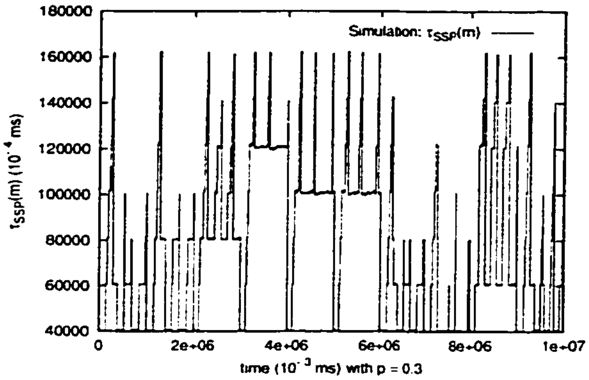
$$\Pr\{VC_i = \text{OFF}\} = \Pr\{R_i > p_i\} = 1 - \int_0^{p_i} 1\, du = 1 - p_i. \tag{4.24}$$

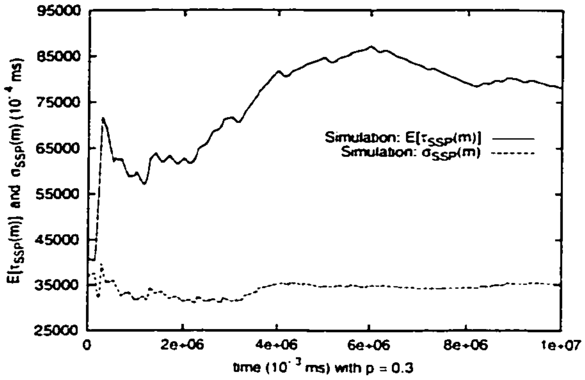
This generates a multicast-tree bottleneck RM-cell RTT observation $\tau(t_k)$ at the end of T_k. Notice that since the 15 random cross-traffic VC_i independently switch between ON and OFF states at any T_k under the control of R_i, there are possibly multiple congested links and paths during some overlapping ON periods. Repeating the above observation procedure in T_k independently for $k = 1, 2, \cdots, N$, we can obtain N multicast-tree bottleneck RM-cell RTT observations $\tau(t_1)$, $\tau(t_2)$, $\cdots$, $\tau(t_N)$. Then, the means $\overline{\tau}$ and standard deviations σ of the multicast-tree bottleneck RM-cell RTT can be estimated through their time-sample averages, $\widehat{\overline{\tau}}(T)$ and $\widehat{\sigma}(T)$, respectively, over the simulation observation time T as follows:

$$\overline{\tau} \approx \widehat{\overline{\tau}}(T) = \frac{1}{T}\int_0^T \tau(t)\, dt \qquad \text{and} \qquad \sigma \approx \widehat{\sigma}(T) = \sqrt{\frac{1}{T}\int_0^T \left[\tau(t) - \widehat{\overline{\tau}}(T)\right]^2 dt} \tag{4.25}$$

where $\tau(t)$ is the running instant multicast-tree bottleneck RM-cell RTT observed at time t through the simulation.

Setting $p_i = p$, $T = 10000$ ms, and $N = 100$, Figure 4.7(a) plots the simulated time-sample of the multicast-tree bottleneck RTT of SSP for $p = 0.3$ over T. Figure 4.7(b) plots the running sample-average $\widehat{\overline{\tau}}_{SSP}(t)$ and standard deviation $\widehat{\sigma}_{SSP}(t)$ of the multicast-tree bottleneck RTT, which are obtained through time-averaging. The ending-values of $\widehat{\overline{\tau}}_{SSP}(T)$ and $\widehat{\sigma}_{SSP}(T)$ over the entire simulation observation time T give the time av-





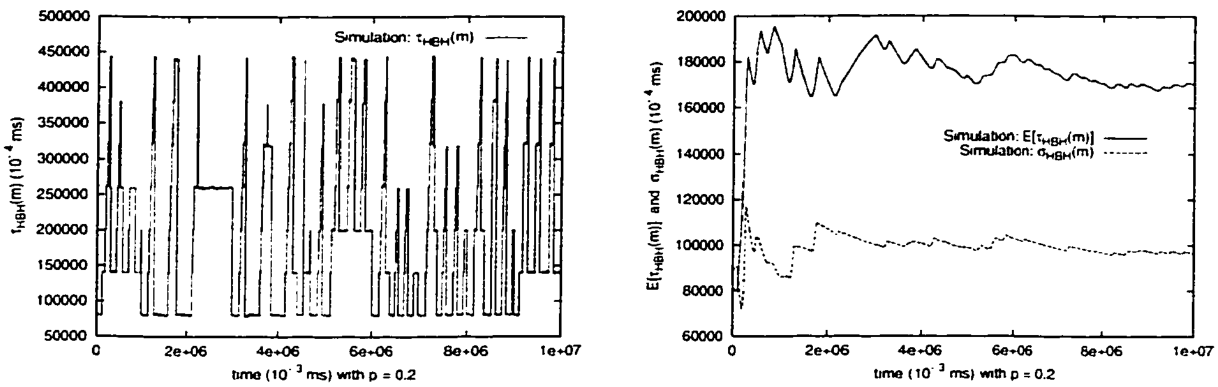
(a) Instant multicast-tree delay $\tau_{SSP}(t)$ vs. t (b) Statistics $\overline{\tau}_{SSP}(t)$ and $\sigma_{SSP}(t)$ vs. t

Figure 4.7: The simulated multicast-tree bottleneck RM-cell RTTs and their statistics for SSP

erage of $\widehat{\overline{\tau}}_{SSP}(T) \approx \overline{\tau}_{SSP}(m)$ and $\widehat{\sigma}_{SSP}(T) \approx \sigma_{SSP}(m)$ for $p = 0.3$. As shown in Figure 4.7(a), the dynamics of $\tau_{SSP}(t)$ evolves randomly, depending on the probability distributions of the cross-traffic sessions over T. However, $\tau_{SSP}(t)$ is always bounded from above by $\tau_{max} = 2m = 16$ ms,[3] and from below by $\tau_{min}^{Limit}(1) = 4$ ms, which confirms Theorem 3.4.2 (also see Table 4.1). Figure 4.7(b) shows that $\widehat{\overline{\tau}}_{SSP}(T) \to 7.78$ ms and $\widehat{\sigma}_{SSP}(T) \to 3.51$ ms, approximately converging to the statistical averages $\overline{\tau}_{SSP}(m)$ and $\sigma_{SSP}(m)$ as $t \to T$ for $p = 0.3$ (see Figures 4.9 and 4.10), respectively. Figures 4.8(a) and (b) present the corresponding simulation results with $p = 0.2$ for HBH. The dynamics of $\tau_{HBH}(t)$ are also found to evolve randomly, following the probability distributions of the cross-traffic sessions over the observation period T. However, $\tau_{HBH}(t)$ is bounded from above by $\tau_{max} = 2m = 44$ ms,[4] and from below by $\tau_{min}^{Limit}(1) = 8$ ms, verifying Theorem 3.4.1 (also see Table 4.1). Figure 4.8(b) shows that $\widehat{\overline{\tau}}_{HBH}(T) \to 17.0$ ms and $\widehat{\sigma}_{HBH}(T) \to 9.7$ ms, converging approximately to the statistical averages $\overline{\tau}_{HBH}(m)$ and $\sigma_{HBH}(m)$ as $t \to T$ for $p = 0.2$ (see Figures 4.9 and 4.10), respectively.

Figures 4.9 and 4.10 plot the simulated means and standard deviations for multicast-tree bottleneck RM-cell RTT against the link-marking probabilities p, and compare them with

[3] The slight exceed of $\tau_{SSP}(t)$ upper-bound over 16 ms as shown in Figure 4.7(a) is due to switching processing delays.
[4] The slight exceed of $\tau_{HBH}(t)$ upper-bound over 44 ms as shown in Figure 4.8(a) is due to switching processing delays.



(a) Instant multicast-tree delay $\tau_{HBH}(t)$ vs. t (b) Statistics $\overline{\tau}_{HBH}(t)$ and $\sigma_{HBH}(t)$ vs. t

Figure 4.8: The simulated multicast-tree bottleneck RM-cell RTTs and their statistics for HBH.

the analytical results derived for SSP and HBH, respectively. Comparing Figure 4.9 and Table 4.1, one can observe that the statistical averages of multicast-tree bottleneck RTTs for both SSP and HBH are generally smaller than the upper bound of the deterministic RTT along each path. This is because the probability distribution $\psi(P_k, p, k)$ of the dominant bottleneck path favors the path P_{k^*} ($k^* \approx 3$) which is closer to P_1 than P_7 or P_8. The simulation results also show the existence of the respective unique p^* that maximizes $\overline{\tau}_{SSP}(m)$ and $\overline{\tau}_{HBH}(m)$, respectively, verifying that $\psi(P_k, p, k)$ has the unique maximum w.r.t (with respect to) p for each path P_k as shown in Theorem 4.4.2.

Figures 4.9 and 4.10 also show that under SSP the average multicast-tree bottleneck RTT over p is always smaller than the upper-bound of $\tau_{min}^{Limit}(j)$ as shown in Table 4.1. In contrast, under HBH, the average multicast-tree bottleneck RTT can be larger than the upper-bound of the physically limiting minimum $\tau_{min}^{Limit}(j)$ for certain values of p. The statistics collected from the simulation show that, on average, the signaling delay for SSP is only about one half of that for HBH as illustrated in Figure 4.9. This advantage remains unchanged when the entire traffic congestion level is varied in the simulated range of p. Hence, the multicast flow control based on SSP is more responsive, and thus more efficient, than HBH. Furthermore, Figure 4.10 shows that the variation of multicast signaling delay

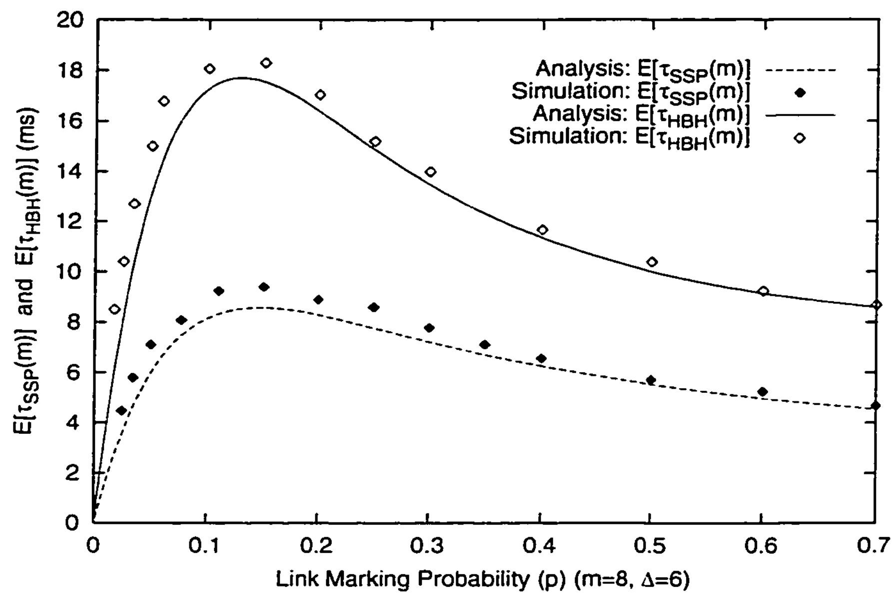


Figure 4.9: Comparison of the simulated RTT delay means with the analytical results: $\overline{\tau}_{SSP}$ and $\overline{\tau}_{HBH}$ vs. p

for SSP is much smaller than that for HBH. So, SSP is more stable than HBH in terms of the multicast signaling delay. In addition, Figures 4.9 and 4.10 also show that the simulation results agree well with the analytical results, thus verifying the correctness of modeling and analytical results for both the deterministic analysis derived in Section 3.3 and the statistical analysis derived in Section 4.2.

4.7 Conclusion

We proposed statistical modeling approaches to analysis of the performance of a class of multicast feedback-synchronization signaling algorithms. Specifically, we developed a independent random model to characterize the multicast signaling delay when the congestion markings of different links are based on RED or REM-like multicast and unicast flow control scheme, where the random markings at different links are independent. Using this model, we derived general expressions for the probability distributions of individual paths in

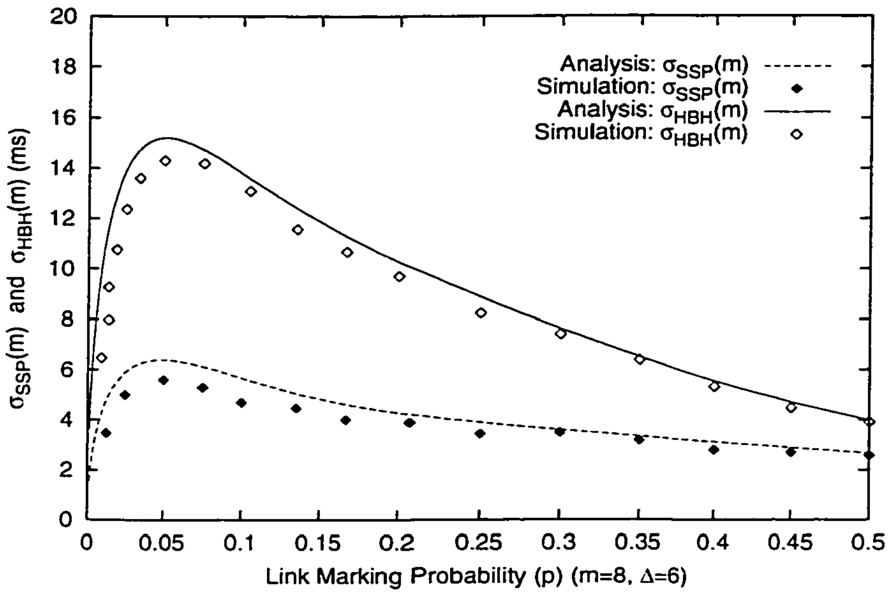


Figure 4.10: Comparison of the simulated standard deviations of RTT with the analytical results: σ_{SSP} and σ_{SSP} vs. p

a multicast tree being the bottleneck. Using the developed statistical model, we derived the first and second moments of a multicast-signaling delay for both HBH and SSP schemes, respectively, when link-markings are independent. The analytical results have also been confirmed by simulations.

CHAPTER 5

MARKOV-CHAIN MODELING FOR THE MULTICAST SIGNALING DELAY ANALYSIS

5.1 Introduction

The independent-marking statistical model developed in Chapter 4 for multicast singling delay analysis builds on the recently-proposed Random Early Marking (REM) [35, 37, 38, 40, 43, 44] and the widely-cited Random Early Detection (RED) [36] flow-control schemes. [1] The independent-marking statistical model is suitable for signaling delay analysis for multicast flow control based on REM- or RED-like schemes, where link-markings are assumed to be independent at different links/routers. However, there are also cases where link-markings may not be independent. In many real unicast or multicast networks, these kinds of cases are even more likely to occur. In such a case, the independent-marking algorithm, modeling, and analysis can only offer approximate results, and their performance and accuracy will be affected by the "degree of dependency" between link-markings. In this chapter, we address the more generalized case of *dependent* link congestion markings. Including dependence in the analysis is usually much more difficult than with the independent-marking assumption.

We develop a Markov-chain model over the link marking/congestion states at different levels in a multicast tree, and a Markov-chain dependency-degree model which can capture

[1] The analytical technique developed in this chapter is also applicable to cases where a link's random congestion state is caused by the flow-control schemes other than REM and RED.

all possible Markov-chain dependency degrees between different link congestion markings. Using the Markov chain and Markov-chain dependency-degree models, we derive the probability distribution for a path to be the multicast-tree bottleneck. We also derive the first and second moments of a multicast signaling delay.

The benefits of our modeling and evaluation technique are two-fold. First, the technique enables a direct quantitative comparison of feedback-synchronization delays between different multicast signaling schemes. Second, the proposed technique establishes a general framework for evaluating the signaling delay of feedback-synchronization-based multicast flow-control algorithms. Although our evaluation focuses on ATM ABR multicast flow control, it can also be applied to any feedback-synchronization-based multicast flow-control algorithm, and to other Markov-chain model based analyses.

The chapter is organized as follows. In Section 5.2, we develop the Markov-chain model and apply it to derive the multicast signaling delay probability distribution for the case of dependent marking. Section 5.3 proposes a Markov-chain dependency-degree model to measure and calculate the one-step transition probabilities. In Section 5.4, we derive expressions for calculating the statistical characteristics of multicast signaling delays. Section 5.5 explores the asymptotical behavior of the derived link-marking Markov chain and its dependency-degree models. Section 5.6 presents the numerical analyses and evaluations, and simulation results to confirm the analytical analyses and findings. The chapter concludes with Section 5.7.

5.2 The Markov Model for Dependent Congestion Markings

In random-marking schemes like REM/RED, and any other flow-control schemes, the marking/congestion state of a link is a function of its queue length. However, the queue lengths of different links carrying the same flows are generally not independent of each other. For instance, if a large (small) queue is built up at a congested upstream link in a multicast tree, the downstream links carrying the same flows are more likely to have large

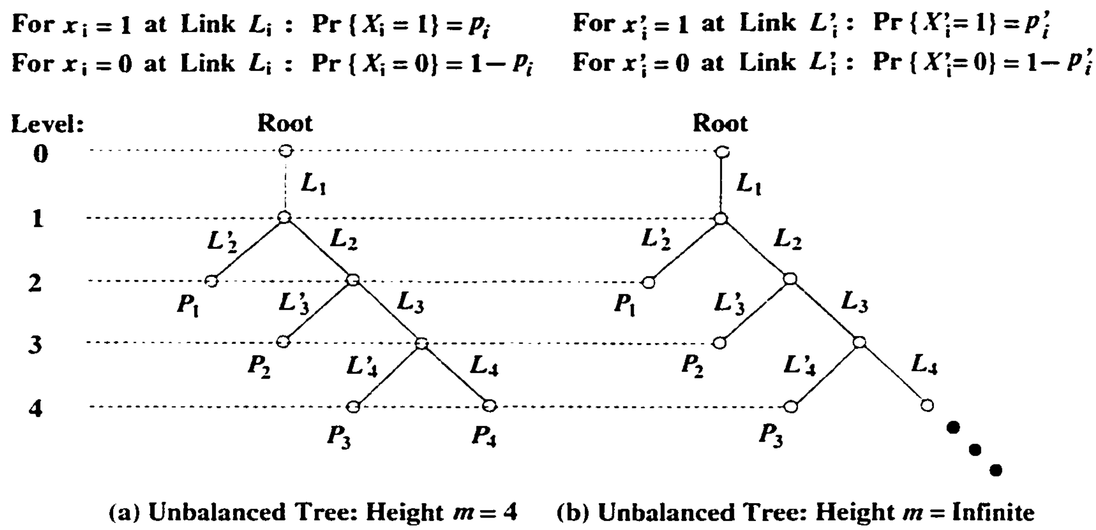


Figure 5.1: Dependent random-marking unbalanced binary-tree model.

(small) queues.

For multicast flow control with *dependent* marking probabilities, we develop a Markov-chain model and a Markov-chain dependency-degree model for measuring and evaluating the degree of the Markov-chain dependency, in order to study the various statistical characteristics of multicast feedback-synchronization delay. The proposed modeling technique can not only be used to analyze the RTT delay of multicast feedback-synchronization signaling, but is also applicable to the general algorithm design/analysis for *both* multicast and unicast flow control.

5.2.1 The Dependent Statistical Model

To analyze the multicast feedback-synchronization signaling with dependent marking probabilities, we introduce the following definition.

Definition 5.2.1 *A dependent random-marking unbalanced binary-tree of height m consists of a set, $\mathcal{L}$, of links which satisfy the following conditions:*

C1. *All links in $\mathcal{L}$ are labeled as shown in Figure 5.1(a) for $m < \infty$ and Figure 5.1(b) for*

$m \to \infty$, respectively, such that

$$\mathcal{L} \triangleq \begin{cases} \{L_1, L_2', L_2, L_3', L_3, \cdots, L_m', L_m\}, & \text{if } m < \infty; \\ \{L_1, L_2', L_2, L_3', L_3, \cdots, L_\infty', L_\infty\}, & \text{if } m \to \infty. \end{cases} \tag{5.1}$$

The link set $\mathcal{L}$ contains m paths, $P_1, P_2, \cdots, P_m$, each of which is represented by its component links as:

$$\begin{cases} P_k \triangleq \{L_1, L_2, \cdots, L_k, L_{k+1}'\}, & \text{if } 1 \le k \le m-1; \\ P_m \triangleq \{L_1, L_2, \cdots, L_m\}, & \text{if } k = m. \end{cases} \tag{5.2}$$

We define P_m as the ***main-stream path*** *which takes only right branches at all branch nodes, and define each P_k, for $1 \le k \le m-1$, as a* ***branch-stream path*** *which consists of k right branches and one left branch at the last branch node (see Figure 5.1). Links L_i and L_i', $\forall i \ge 2$, are at the same level of the multicast tree.*

C2. *The marking state of link L_i (L_i') ($i = 1, 2, \cdots$) is represented by a random variable X_i (X_i') which takes value in $\{0, 1\}$ such that (see the top part of Figure 5.1)*

$$\Pr\{X_i = x_i\} = \begin{cases} p_i, & \text{for } x_i = 1; \\ 1 - p_i, & \text{for } x_i = 0; \end{cases} \qquad \Pr\{X_i' = x_i'\} = \begin{cases} p_i', & \text{for } x_i' = 1; \\ 1 - p_i', & \text{for } x_i' = 0; \end{cases} \tag{5.3}$$

where p_i (p_i') is the marking probability for L_i (L_i') and is determined by

$$p_i = \begin{cases} 1 - \phi^{-\gamma \overline{q}_i}, & \text{if REM is used;} \\ \left[\dfrac{\overline{q}_i - min_{th}}{max_{th} - min_{th}}\right] p_{max}, & \text{if RED is used;} \end{cases} \tag{5.4}$$

$$p_i' = \begin{cases} 1 - \phi^{-\gamma \overline{q'}_i}, & \text{if REM is used;} \\ \left[\dfrac{\overline{q'}_i - min_{th}}{max_{th} - min_{th}}\right] p_{max}, & \text{if RED is used;} \end{cases} \tag{5.5}$$

where $0 < p_i, (p_i') < 1$ (since p_i (p_i') reflects the degree of traffic load, we will use the terms "marking probability" and "traffic load" interchangeably for p_i (p_i')); $\overline{q}_i$ ($\overline{q'}_i$) is the average queue size at L_i (L_i'); $\gamma > 0$ is the step size; $\phi > 1$ for REM; p_{max} is the maximum marking probability; max_{th} (min_{th}) is the high (low) queue threshold for RED;

C3. *The congestion marking states at all links are dependent, and satisfy the Markovian property such that*

$$\Pr\{X_i = x_i \mid X_{i-1} = x_{i-1}, X'_{i-1} = x'_{i-1}, X_{i-2} = x_{i-2}, X'_{i-2} = x'_{i-2}, \cdots, X_1 = x_1\}$$
$$= \Pr\{X_i = x_i \mid X_{i-1} = x_{i-1}\}; \tag{5.6}$$
$$\Pr\{X'_i = x'_i \mid X_{i-1} = x_{i-1}, X'_{i-1} = x'_{i-1}, X_{i-2} = x_{i-2}, X'_{i-2} = x'_{i-2}, \cdots, X_1 = x_1\}$$
$$= \Pr\{X'_i = x'_i \mid X_{i-1} = x_{i-1}\}; \tag{5.7}$$

C4. *The congestion marking states within the same level are also dependent and satisfy the following properties:*

$$\Pr\{X_i = x_i \mid X'_i = x'_i, X_{i-1} = x_{i-1}, X'_{i-1} = x'_{i-1}, X_{i-2} = x_{i-2}, X'_{i-2} = x'_{i-2}, \cdots, X_1 = x_1\}$$
$$= \Pr\{X_i = x_i \mid X_{i-1} = x_{i-1}\} \tag{5.8}$$
$$\Pr\{X'_i = x'_i \mid X_i = x_i, X_{i-1} = x_{i-1}, X'_{i-1} = x'_{i-1}, X_{i-2} = x_{i-2}, X'_{i-2} = x'_{i-2}, \cdots, X_1 = x_1\}$$
$$= \Pr\{X'_i = x'_i \mid X_{i-1} = x_{i-1}\}; \tag{5.9}$$

■

Remarks on Definition 5.2.1 (C3 and C4): We only consider the *upstream* and *same-level dependence* of link marking states as described by Eqs. (5.6), (5.7), (5.8), and (5.9), because the multicast-tree signaling delay analysis to be developed below need not consider the *downstream* dependence. The congestion information on the links above the immediate-next upstream link or on the link at the same level (see C4) is all concentrated into, and carried over by, the given congestion information on the immediate-next upstream link. Conditions **C3** and **C4** are reasonable because one link's congestion state depends most on its immediate upstream link's congestion state. The upstream's influence on a downstream link's congestion state propagates through its immediate upstream link which carries same flows, and thus, as long as the immediate upstream link's congestion state is given, the probability distribution at the downstream link is independent of the congestion state at links which are located above the immediate upstream link or at the same level as indicated by conditions **C3** and **C4** in Eqs. (5.6) through (5.9).

```
On receipt of a feedback RM cell from the i-th branch:
  if (conn_patt_vec(i) ≠ 1) { ! Only process connected downstream branches;
    resp_branch_vec(i) := 1; ! Mark the connected and responsive branch;
    MCI := MCI ∨ CI; ! CI (Congestion Indicator) is randomly marked at router; MCI is consolidated CI
    MER := min{MER, ER}; ! ER (Explicit Rate) information processing; MER is the consolidated ER
    if (conn_patt_vec ⊕ resp_branch_vec = 1) { ! This is the "Soft Synchronization" operation
      send RM cell (dir := back, ER := MER, CI := MCI); ! Send a fully-consolidated RM cell upstream
      no_resp_timer := N_nrt; ! Reset the non-responsive timer for non-responsive branch detection/removal
      resp_branch_vec := 0); ! Reset the responsive branch vector
      MCI := 0; MER := ER;}}; ! Reset RM-cell control variable.
```

Figure 5.2: Pseudocode for the Soft Synchronization Protocol (SSP).

5.2.2 Probability Distribution of the Dominant Bottleneck Path

To ensure reliable data transmission, the multicast ABR service needs to adjust the source rate to the minimum available bandwidth share that can be supported by the most congested path. Clearly, based on the OR rule (see the multicast signaling algorithms detailed in Figure 5.2 and [45]. Specifically, at the heart of SSP [6, 11, 45] is a pair of connection-update vectors: (i) the connection pattern vector, *conn_patt_vec*, where *conn_patt_vec*$(i) = 0(1)$ indicates the i-th output port of the switch is (not) a downstream branch of the multicast connection. Thus, *conn_patt_vec*$(i) = 0(1)$ implies that a data cell should (not) be sent to the i-th downstream branch and a feedback RM cell is (not) expected from the i-th downstream branch;[2] (ii) the responsive branch vector, *resp_branch_vec*, is initialized to $\underline{0}$ and reset to $\underline{0}$ whenever a consolidated RM cell is sent upward from the switch. *resp_branch_vec*(i) is set to 1 if a feedback RM cell is received from the i-th downstream branch. A pseudocode [11, 45] of the switch RM-cell processing algorithm is given in Figure 5.2. On receipt of a returned feedback RM-cell, the switch first marks its corresponding bit in the *resp_branch_vec* and then conducts RM-cell consolidation operations. If the modulo-2 addition (the soft-synchronization operation), *conn_patt_vec* $\oplus$ *resp_branch_vec*

[2]Note that the negative logic is used for convenience of implementation.

equals $\underline{1}$, an all 1's vector, indicating all feedback RM cells synchronized, then a fully-consolidated feedback RM cell is generated and sent upward. But, if the modulo-2 addition is not equal to $\underline{1}$, the switch needs to await other feedback RM cells for synchronization. Notice that since the synchronization algorithm allows feedback RM cells corresponding to different forward RM cells to be consolidated, the feedback RM cells are "softly-synchronized" at branch nodes.), the *shortest* bottleneck path in a multicast tree dominates the source's flow-control decisions and the RTT of flow-control feedback loop. To explicitly model this feature, we introduce the following definition.

Definition 5.2.2 *Among all concurrent bottleneck paths in a multicast tree, the bottleneck path of **minimum** length is called the **dominant bottleneck path** (also called **multicast-tree bottleneck path**), and its RM-cell RTT is called the **multicast-tree bottleneck RM-cell RTT** or simply **multicast-tree RTT**.*

Based on Definitions 5.2.1 and 5.2.2, the following proposition lays a foundation for deriving the distribution of the dominant bottleneck path.

Proposition 5.2.1 *The sequence of random marking states $\{X_1, X_2, \cdots, X_{m-1}, X_m\}$ (for the tree height $m < \infty$ and $m \to \infty$, respectively) in Definition 5.2.1 defines a 2-state discrete-indexed Markov chain over the links on the main-stream path $P_m = \{L_1, L_2, \cdots, L_m\}$, and the sequence of marking states $\{X_1, X_2, \cdots, X_k, X'_{k+1}\}$ in Definition 5.2.1 on each branch-stream path $P_k = \{L_1, L_2, \cdots, L_k, L'_{k+1}\}$, for $k = 1, 2, \cdots, m-1$, also define a 2-state (finite-sequence) Markov chain.*

Proof. The proof follows from conditions **C3** of Definition 5.2.1. ∎

Remarks on Proposition 5.2.1: Unlike the traditional definition of Markov chain/process where the random-variable sequence index set is time, we define the Markov chain for every path (including the main- and branch-stream paths) which is indexed by the (discrete) link sequence number associated with that path.

Since the mathematical properties/treatments and random marking definitions for both the Markov chain defined over the main-stream path P_m and the Markov chain defined over the branch-stream paths P_k ($k = 1, 2, \cdots, m-1$) are the same, except that the last link's marking state differs in labeling by a "$'$" symbol (see Proposition 5.2.1), we will henceforth use $\{X_i\}$ to represent the Markov chain defined over both the main- and branch-stream paths, and explicitly state otherwise.

Applying Proposition 5.2.1, the theorem given below derives the probability distributions of the dominant-bottleneck path.

Theorem 5.2.1 *If a dependent-marking multicast tree of height m as defined in Definition 5.2.1 is flow-controlled under SSP or HBH, then the following claims hold:*

<u>Claim 1:</u> *If $m \to \infty$, then there exists **one and only one** dominant bottleneck path, and the probability distribution, $\psi_d(P_k, \infty)$, that P_k becomes the dominant bottleneck path, is*

$$\psi_d(P_k,\infty) = \begin{cases} 1 - \Pr\{X_1 = 0\}\Pr\{X_2' = 0 \mid X_1 = 0\}, & \text{if } k = 1; \\ \\ \Pr\{X_1 = 0\}\Pr\{X_k' = 0 \mid X_{k-1} = 0\}\Big[\Pr\{X_k = 1 \mid X_{k-1} = 0\} \\ \quad +\Pr\{X_k = 0 \mid X_{k-1} = 0\}\Pr\{X_{k+1}' = 1 \mid X_k = 0)\Big] \\ \quad \cdot \prod_{i=1}^{k-2}\Big\{\Pr\{X_{i+1} = 0 \mid X_i = 0\}\Pr\{X_{i+1}' = 0 \mid X_i = 0\}\Big\}, & \text{if } k \geq 2; \end{cases} \tag{5.10}$$

The $\psi_d(P_k, \infty)$ given in Eq. (5.10) satisfies the following normalization condition:

$$\lim_{m\to\infty} \sum_{k=1}^{m} \psi_d(P_k, \infty) = 1; \tag{5.11}$$

<u>Claim 2:</u> *If $m < \infty$, then there exists **at most one** dominant bottleneck path, and the probability distribution, $\psi_d(P_k, m)$, that P_k becomes the dominant bottleneck path, is*

given by

$$\psi_d(P_k, m) = \begin{cases} 1 - \Pr\{X_1 = 0\}\Pr\{X_2' = 0 \mid X_1 = 0\}, & \text{if } k = 1; \\ \\ \Pr\{X_1 = 0\}\Pr\{X_k' = 0 \mid X_{k-1} = 0\}\Big[\Pr\{X_k = 1 \mid X_{k-1} = 0\} & \\ \quad +\Pr\{X_k = 0 \mid X_{k-1} = 0\}\Pr\{X_{k+1}' = 1 \mid X_k = 0)\Big] & \\ \quad \cdot \prod_{i=1}^{k-2}\Big\{\Pr\{X_{i+1} = 0 \mid X_i = 0\}\Pr\{X_{i+1}' = 0 \mid X_i = 0\}\Big\}, & \text{if } k \geq 2; \\ \\ \Pr\{X_1 = 0\}\Pr\{X_m = 1 \mid X_{m-1} = 0\}\Pr\{X_m' = 0 \mid X_{m-1} = 0\} & \\ \quad \cdot \prod_{i=1}^{m-2}\Big\{\Pr\{X_{i+1} = 0 \mid X_i = 0\}\Pr\{X_{i+1}' = 0 \mid X_i = 0\}\Big\}, & \text{if } k = m; \end{cases} \tag{5.12}$$

Proof. The proof is provided in Appendix T. ■

Remarks on Theorem 5.2.1: We observe that by Eq. (5.10), $\lim_{k\to\infty} \psi_d(P_k, \infty) = 0$. This is expected, since a longer bottleneck path is always dominated by a co-existing shorter bottleneck path, if any. Thus, when $k \to \infty$ as $m \to \infty$, P_∞ is always dominated by a shorter bottleneck path for $0 < p_i, p_i' < 1$, $i = 1, 2, \cdots, \infty$. That is, $\psi_d(P_\infty, \infty) = 0$. In addition, notice that by Eq. (5.11) we have $\sum_{k=1}^{\infty} \psi_d(P_k, \infty) = 1$, which also makes sense because as the unbalanced-tree's height $m \to \infty$ and $0 < p_i, p_i' < 1$, there always exists (with probability 1) one and only one dominant bottleneck path in a multicast tree. On the other hand, for the case of $m < \infty$, by Eqs. (5.12) and (5.11) we have $\sum_{k=1}^{m} \psi_d(P_k, m) \leq 1$, implying the possibility that there is no dominant bottleneck path in the multicast tree of height $m < \infty$. This is also expected because $0 < p_i, p_i' < 1$.

5.3 Modeling of Markov-Chain Dependency Degree

To use Eqs. (5.10) and (5.12), we need to derive explicit expressions for $\Pr\{X_i = x_i \mid X_{i-1} = x_{i-1}\}$ and $\Pr\{X_i' = x_i' \mid X_{i-1} = x_{i-1}\}$, which are the fundamental conditional distribution functions used in Eqs. (5.10) and (5.12). However, it is difficult to know/compute the accurate dependency between two random variables. To solve this problem, we propose to use a real-valued Markov-chain *dependency-degree factor* $\alpha \in [0, 1]$ to quantify all possible degrees of dependency between the random variables in the Markov chain's one-step transition probabilities. Using this dependency-degree factor, one can evaluate the Markov chain's any possible degree of dependency ranging from "independent" to "perfectly dependent", without knowing *a priori* the dependency-degree of the two random variables.

In general, two dependent random events can affect each other either *positively* or *negatively.* For instance, if occurrence of one event is likely to trigger another, then they are said to be *positively-dependent*. On the other hand, if occurrence of one event makes another event unlikely to occur, then they are said to be *negatively-dependent*. As we discussed earlier, an upstream link's being congested (uncongested) state will make the downstream links carrying the same flows more likely (unlikely) to be congested. So, the positive dependence can accurately characterize the dependence of link markings. To quantitatively describe this feature, we introduce the following definition:

Definition 5.3.1 *Two dependent link marking states X_i and X_{i+1} are said to be positively (negatively) dependent if* $\Pr\{X_{i+1} = x \mid X_i = x\} > \Pr\{X_{i+1} = x \mid X_i = \overline{x}\}$ ($\Pr\{X_{i+1} = x \mid X_i = x\} < \Pr\{X_{i+1} = x \mid X_i = \overline{x}\}$), *where* $x \in \{0, 1\}$.

Based on Definition 5.3.1, the theorem given below models the dependency-degree between the random variables of the Markov chain. Notice that the theorem below only gives the results for the case of $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$ and $\Pr\{X_i = 1\} = p_i$, and it also holds for the case of $\Pr\{X'_{i+1} = x'_{i+1} \mid X_i = x_i\}$ and $\Pr\{X'_i = 1\} = p'_i$ with the similar results that we omitted.

Theorem 5.3.1 *Consider the Markov chain $\{X_i\}$ defined on link marking states on every path (for both main-stream and branch-stream) in the multicast tree specified by Definition 5.2.1. If $\{X_i\}$ is* ***positively dependent****, and the link marking-probability is equal to* $\Pr\{X_i = 1\} = p_i$, *then the following claims hold.*

<u>**Claim 1.**</u> *The conditional distribution* $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$, *with* $x_i, x_{i+1} \in \{0, 1\}$,

is upper- and lower-bounded by

$$1-p_{i+1} \le \Pr\{X_{i+1}=0 \mid X_i=0\} \le \begin{cases} 1, & \text{if } p_i \ge p_{i+1}; \\ \dfrac{1-p_{i+1}}{1-p_i}, & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.13}$$

$$p_{i+1} \ge \Pr\{X_{i+1}=1 \mid X_i=0\} \ge \begin{cases} 0, & \text{if } p_i \ge p_{i+1}; \\ \dfrac{p_{i+1}-p_i}{1-p_i}, & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.14}$$

$$1-p_{i+1} \ge \Pr\{X_{i+1}=0 \mid X_i=1\} \ge \begin{cases} \dfrac{p_i-p_{i+1}}{p_i}, & \text{if } p_i \ge p_{i+1}; \\ 0, & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.15}$$

$$p_{i+1} \le \Pr\{X_{i+1}=1 \mid X_i=1\} \le \begin{cases} \dfrac{p_{i+1}}{p_i}, & \text{if } p_i \ge p_{i+1}; \\ 1, & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.16}$$

Claim 2. $\exists\, \alpha_i(\alpha_i') \in [0,1]$ *such that all possible dependency-degrees between* X_i *and* X_{i+1} *(*X_{i+1}'*) can be measured by the real-valued Markov-chain* ***dependency-degree factor*** α_i *(*α_i'*), and* [3]

$$\begin{cases} \alpha_i = 0 \ \textbf{ iff } \ X_i \textit{ and } X_{i+1} \textit{ are independent;} \\ \alpha_i = 1 \ \textbf{ iff } \ X_i \textit{ and } X_{i+1} \textit{ are } \textbf{\textit{perfectly}} \textit{ dependent;} \end{cases} \tag{5.17}$$

and

$$\begin{cases} \alpha_i' = 0 \ \textbf{ iff } \ X_i \textit{ and } X_{i+1}' \textit{ are independent;} \\ \alpha_i' = 1 \ \textbf{ iff } \ X_i \textit{ and } X_{i+1}' \textit{ are } \textbf{\textit{perfectly}} \textit{ dependent;} \end{cases} \tag{5.18}$$

Claim 3. *The conditional distributions* $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$, *with* $x_i, x_{i+1} \in \{0,1\}$,

[3] Examples of the *perfectly* dependent events discussed below include that two events are identical or one event is a sub-event of the other.

are determined by

$$\Pr\{X_{i+1}=0 \mid X_i=0\} = \begin{cases} 1-(1-\alpha_i)\,p_{i+1}, & \text{if } p_i \geq p_{i+1}; \\ (1-\alpha_i)(1-p_{i+1}) + \alpha_i\left(\dfrac{1-p_{i+1}}{1-p_i}\right), & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.19}$$

$$\Pr\{X_{i+1}=1 \mid X_i=0\} = \begin{cases} (1-\alpha_i)\,p_{i+1}, & \text{if } p_i \geq p_{i+1}; \\ (1-\alpha_i)\,p_{i+1} + \alpha_i\left(\dfrac{p_{i+1}-p_i}{1-p_i}\right), & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.20}$$

$$\Pr\{X_{i+1}=0 \mid X_i=1\} = \begin{cases} (1-\alpha_i)(1-p_{i+1}) + \alpha_i\left(\dfrac{p_i-p_{i+1}}{p_i}\right), & \text{if } p_i \geq p_{i+1}; \\ (1-\alpha_i)(1-p_{i+1}), & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.21}$$

$$\Pr\{X_{i+1}=1 \mid X_i=1\} = \begin{cases} (1-\alpha_i)p_{i+1} + \alpha_i\left(\dfrac{p_{i+1}}{p_i}\right), & \text{if } p_i \geq p_{i+1}; \\ p_{i+1} + \alpha_i(1-p_{i+1}), & \text{if } p_i < p_{i+1}; \end{cases} \tag{5.22}$$

where $i = 1, 2, \cdots$, and α_i is the dependency-degree factor defined in Claim 2 of Theorem 5.3.1.

Proof. See Appendix U. ∎

Remarks on Theorem 5.3.1: Claim 1 finds the upper and lower bounds of all 4 possible 2-state Markov chain one-step transition probabilities as functions of the marginal link-marking probabilities p_i and p_{i+1} specified by networks. Claim 2 ensures the existence of a real-valued dependence-degree factor $\alpha_i \in [0, 1]$. It also proves the completeness of the Markov-chain dependence-degree factor by mapping all possible degrees of dependency onto the real-valued interval $[0, 1]$. Claim 3 derives expressions for all 4 possible 2-state Markov chain one-step transition probabilities, expressing the conditional distributions as the functions of their marginal distributions.

Applying Theorem 5.3.1 and Eqs. (5.19)–(5.22) to Theorem 5.2.1, we obtain the general-case (heterogeneous) expressions for calculating the multicast bottleneck path probability distributions as follows.

Corollary 5.3.1 *Let a dependent-marking multicast tree of height m as defined in Definition 5.2.1 be flow-controlled under SSP or HBH. If the one-step transition probability* *of the Markov chain $\{X_i\}$ defined over every path (including the main- and branch-stream paths) is specified by the dependency-factor vector $\vec{\alpha} \triangleq (\alpha_1, \alpha'_1, \alpha_2, \alpha'_2, \alpha_3, \alpha'_3, \cdots)$ which is derived in Theorem 5.3.1, and further, denote the link marking probability vector by $\vec{p} \triangleq (p1, p'_1, p_2, p'_2, p_3, p'_3, \cdots)$, respectively, then the following claims hold.*

<u>**Claim 1:**</u> *If $m \to \infty$, then there exists one and only one dominant bottleneck path, and the probability distribution, denoted by $\psi_d(P_k, \vec{\alpha}, \vec{p}, \infty)$, that P_k becomes the dominant bottleneck path, is determined by*

$$\psi_d(P_k, \vec{\alpha}, \vec{p}, \infty) = \begin{cases} 1 - (1-p_1)\left[1 - (1-\alpha'_1)p'_2\right], & \text{if } k = 1; \\ (1-p_1)\left[1 - (1-\alpha'_{k-1})p'_k\right]\Big[(1-\alpha_{k-1})p_k + \left[1 - (1-\alpha_{k-1})p_k\right] \\ \quad \cdot (1-\alpha'_k)p'_{k+1}\Big] \prod_{i=1}^{k-2}\Big\{\left[1-(1-\alpha_i)p_{i+1}\right]\left[1-(1-\alpha'_i)p'_{i+1}\right]\Big\}, & \text{if } k \geq 2; \end{cases} \tag{5.23}$$

and, $\psi_d(P_k, \vec{\alpha}, \vec{p}, \infty)$ given in Eq. (5.23) satisfies the following normalization condition:

$$\lim_{m \to \infty} \sum_{k=1}^{m} \psi_d(P_k, \vec{\alpha}, \vec{p}, \infty) = 1; \tag{5.24}$$

<u>**Claim 2:**</u> *If $m < \infty$, then there exists at most one dominant bottleneck path, and the probability distribution, denoted by $\psi_d(P_k, \vec{\alpha}, \vec{p}, m)$, that P_k becomes the dominant bottleneck path, is determined by*

$$\psi_d(P_k, \vec{\alpha}, \vec{p}, m) = \begin{cases} 1 - (1-p_1)\left[1 - (1-\alpha'_1)p'_2\right], & \text{if } k = 1; \\ (1-p_1)\left[1 - (1-\alpha'_{k-1})p'_k\right]\Big[(1-\alpha_{k-1})p_k + \left[1 - (1-\alpha_{k-1})p_k\right] \\ \quad \cdot (1-\alpha'_k)p'_{k+1}\Big] \prod_{i=1}^{k-2}\Big\{\left[1-(1-\alpha_i)p_{i+1}\right]\left[1-(1-\alpha'_i)p'_{i+1}\right]\Big\}, & \text{if } k \geq 2; \\ (1-p_1)(1-\alpha_{m-1})p_m\left[1-(1-\alpha'_{m-1})p'_m\right] \\ \quad \cdot \prod_{i=1}^{m-2}\Big\{\left[1-(1-\alpha_i)p_{i+1}\right]\left[1-(1-\alpha'_i)p'_{i+1}\right]\Big\}, & \text{if } k = m; \end{cases} \tag{5.25}$$

Proof. The proof follows by plugging Eqs. (5.19) through (5.22) of Theorem 5.3.1 into Eqs. (5.10), (5.11) and (5.12) of Theorem 5.2.1. ■

Remarks on Corollary 5.3.1: We can use Eqs. (5.23) and (5.25), and tune up the dependence-degree factor $\vec{\alpha}$ to see how the system performs with different dependence-degrees. More importantly, the completeness of this approach guarantees that the actual unknown Markov-chain dependency degree imposed by the practical problems can always be covered by tuning α_i in the interval [0,1], $\forall i$. Moreover, Eqs. (5.23) and (5.25) provide very general probability distribution expressions since one can arbitrarily select $\vec{\alpha}$ and $\vec{p}$ for different links to handle the heterogeneity. Eqs. (5.23) and (5.25) reduce to the probability distribution expressions of $\psi(P_k, m)$ derived for the multicast signaling delay analysis under *independent* random-marking [45] by letting $\vec{\alpha} = \vec{0}$ (independent), verifying the correctness of Eqs. (5.23) and (5.25).

5.4 Statistical Properties of Multicast Signaling Delays

Using the probability distribution derived in Corollary 5.3.1 and Eqs. (3.1) and (3.6) of $\tau_u(j, \Delta)$ derived in Theorems 3.4.1 and 3.4.2 in Appendix R, the following theorem derives the probability distributions, their properties, and the means and variances of multicast signaling delays under SSP and HBH, respectively, for the homogeneous case and with $m < \infty$.

Theorem 5.4.1 *Let a dependent-marking multicast tree of height m as defined in Definition 5.2.1 be flow-controlled under SSP and HBH, respectively, with the RM-cell interval Δ. If $m < \infty$, $0 < p_i = p_i' = p < 1$ and $0 \leq \alpha_i = \alpha_i' = \alpha \leq 1$, $\forall i$ (the homogeneous case),[4] then the following claims hold:*

<u>Claim 1:</u> *The probability distribution that P_k becomes the dominant bottleneck path, de-*

[4] The analytical results derived from the homogeneous case can be easily extended to the heterogeneous case where p_i and α_i are different $\forall i$.

noted by $\psi_d(P_k, \alpha, p, m)$, *is determined by*

$$\psi_d(P_k, \alpha, p, m) = \begin{cases} 1-(1-p)\left[1-(1-\alpha)p\right], & \text{if } k=1; \\ (1-\alpha)(1-p)p\left[2-(1-\alpha)p\right]\left[1-(1-\alpha)p\right]^{2k-3}, & \text{if } k\geq 2; \\ (1-\alpha)(1-p)p\left[1-(1-\alpha)p\right]^{2m-3}, & \text{if } k=m; \end{cases} \quad (5.26)$$

<u>Claim 2:</u> *For each path* P_k *and a given* α, $\psi_d(P_k, \alpha, p, m)$ *attains the unique maximum at*

$$p^* \triangleq \arg\max_{0<p<1} \psi_d(P_k, \alpha, p, m) = \begin{cases} 1, & \text{if } k=1; \\ \dfrac{m-(m-1)\alpha-\sqrt{[m-(m-1)\alpha]^2-(1-\alpha)(2m-1)}}{(1-\alpha)(2m-1)}, & \text{if } k=m; \end{cases} \quad (5.27)$$

and for $2 \leq k \leq (m-1)$, p^* *is non-negative and no larger than* 1 *real-valued root of the following cubic equation:*

$$2k(1-\alpha)^2p^3 + (1-\alpha)\left[(2k-1)\alpha-6k\right]p^2 - 2\left[(2k-1)\alpha-2k-1\right]p - 2 = 0. \quad (5.28)$$

<u>Claim 3:</u> *For each path* P_k *and a given* p, $\psi_d(P_k, \alpha, p, m)$ *attains the unique maximum at*

$$\alpha^* \triangleq \arg\max_{0<\alpha<1} \psi_d(P_k, \alpha, p, m) = \begin{cases} \dfrac{p-1}{p} + \dfrac{1}{p}\sqrt{1-\dfrac{2}{2k-1}}, & \text{if } 2\leq k\leq m-1 \text{ and } k \geq \left\lceil \dfrac{1}{2} + \dfrac{1}{p(2-p)} \right\rceil; \\ 1-\dfrac{1}{2(m-1)p}, & \text{if } k=m \text{ and } k\geq \left\lceil 1+\dfrac{1}{2p}\right\rceil; \end{cases} \quad (5.29)$$

<u>Claim 4:</u> *If Markov-chain dependency-degree factor* $\alpha = \alpha_0 > 0$ *for a given* α_0, *it shifts the probability distribution of multicast-tree bottleneck path from shorter paths to longer ones. If the tree height* m *satisfies:*

$$m \geq \left\lfloor \frac{\log\sqrt{\dfrac{1}{1-\alpha_0}}}{\log\dfrac{1-(1-\alpha_0)p}{1-p}} + 2.5 \right\rfloor, \quad (5.30)$$

then there exists the unique "dependency-balanced path" $P_{\tilde{k}}$ such that $2 \le \tilde{k} \le m-1$ and

$$\begin{cases} \psi_d(P_k, \alpha, p, m)\,|_{\alpha=0} \ \ge \ \psi_d(P_k, \alpha, p, m)\,|_{\alpha=\alpha_0}, & \text{if } k \le \tilde{k}; \\ \\ \psi_d(P_k, \alpha, p, m)\,|_{\alpha=0} \ < \ \psi_d(P_k, \alpha, p, m)\,|_{\alpha=\alpha_0}, & \text{if } k > \tilde{k}; \end{cases} \tag{5.31}$$

where $\psi_d(P_k, \alpha, p, m)$ is given by Eq. (5.26), and the "dependency-balanced path number" $\tilde{k}$ is determined by

$$\tilde{k} = \left\lfloor \frac{\log \sqrt{\dfrac{2-p}{(1-\alpha_0)[2-(1-\alpha_0)p]}}}{\log \dfrac{1-(1-\alpha_0)p}{1-p}} + 1.5 \right\rfloor; \tag{5.32}$$

<u>**Claim 5:**</u> *The means of multicast-tree bottleneck RM-cell RTT, denoted by $\overline{\tau}_{SSP}(\alpha, p, m)$ and $\overline{\tau}_{HBH}\,(\alpha, p, m)$ for the SSP and HBH schemes, respectively, are determined by:*

$$\begin{aligned} \overline{\tau}_{SSP}(\alpha, p, m) = & \left[p + (1-\alpha)(p-p^2)\right] \left[2m - \left\lfloor \frac{2(m-2)}{\Delta} \right\rfloor \Delta \right] + 2m(1-p)\left[1-(1-\alpha)p\right] \\ & \cdot \left\{ 1 + \left[1-(1-\alpha)p\right]^{2(m-2)} \left[(1-\alpha)p - 1\right] \right\} - (1-\alpha)(1-p)p \\ & \cdot \left[2-(1-\alpha)p\right] \Delta \sum_{k=2}^{m-1} \left\{ \left\lfloor \frac{2(m-k-1)}{\Delta} \right\rfloor \left[1-(1-\alpha)p\right]^{2k-3} \right\}, \end{aligned} \tag{5.33}$$

$$\begin{aligned} \overline{\tau}_{HBH}(\alpha, p, m) = & \frac{(1-p)\Theta(\Delta)}{(1-\alpha)p\left[2-(1-\alpha)p\right]} \Big\{ 2\left[1-(1-\alpha)p\right] - \left[1-(1-\alpha)p\right]^3 \\ & - m\left[1-(1-\alpha)p\right]^{2m-3} + (m-1)\left[1-(1-\alpha)p\right]^{2m-1} \Big\} + (1-p) \\ & \cdot \left[1-(1-\alpha)p\right]^{2m-3} \Big\{ (1-\alpha)p\left[2+(m-1)\Theta(\Delta)\right] - 2 \Big\} + (2+\Theta(\Delta)) \\ & \cdot \left[p + (1-\alpha)(p-p^2)\right] + 2(1-p)\left[1-(1-\alpha)p\right]; \end{aligned} \tag{5.34}$$

where $\Theta(\Delta)$ is defined by Eq. (3.2) in Appendix R;

<u>**Claim 6:**</u> *The variances of multicast-tree bottleneck RM-cell RTT, denoted by $\sigma^2_{SSP}(\alpha, p, m)$*

and σ^2_{HBH} (α, p, m) for the SSP and HBH schemes, respectively, are determined by:

$$\begin{aligned}
\sigma^2_{SSP}(\alpha,p,m) = {} & 4m^2+4m^2(1-p)\,[1-(1-\alpha)p]^{2m-3}\,[(1-\alpha)p-1]-(1-\alpha)(1-p)p \\
& [2-(1-\alpha)p]\left\{4m\Delta\sum_{k=2}^{m-1}\left\{\left\lfloor\frac{2(m-k-1)}{\Delta}\right\rfloor(1-(1-\alpha)p)^{2k-3)}\right\}\right. \\
& \left.-\Delta^2\sum_{k=2}^{m-1}\left\{\left\lfloor\frac{2(m-k-1)}{\Delta}\right\rfloor^2(1-(1-\alpha)p)^{2k-3)}\right\}\right\} \\
& +p\Big[1+(1-\alpha)(1-p)\Big]\left\{\Delta^2\left\lfloor\frac{2(m-2)}{\Delta}\right\rfloor^2-4m\Delta\right. \\
& \left.\cdot\left\lfloor\frac{2(m-2)}{\Delta}\right\rfloor\right\}-\overline{\tau}^2_{SSP}(\alpha,p,m), \qquad (5.35)
\end{aligned}$$

$$\begin{aligned}
\sigma^2_{HBH}(\alpha,p,m) = {} & [1+(1-\alpha)(1-p)]\,p(2+\Theta(\Delta))^2+(1-\alpha)(1-p)p \\
& \cdot[1-(1-\alpha)p]^{2m-3}\,[2+(m-1)\Theta(\Delta)]^2+4(1-p)\,[1-(1-\alpha)p] \\
& \cdot\left\{1-[1-(1-\alpha)p]^{2(m-2)}\right\}+\frac{4(1-p)\,[1-(1-\alpha)p]\,\Theta(\Delta)}{(1-\alpha)p\,[2-(1-\alpha)p]} \\
& \cdot\left\{2-[1-(1-\alpha)p]^2-m\,[1-(1-\alpha)p]^{2(m-2)}+(m-1)\right. \\
& \left.\cdot[1-(1-\alpha)p]^{2(m-1)}\right\}+\frac{(1-p)\Theta^2(\Delta)}{(1-\alpha)^2\,[2-(1-\alpha)p]^2\,[p^2-(1-\alpha)p^3]} \\
& \cdot\left\{1+[1-(1-\alpha)p]^2-[2-(1-\alpha)p]^3\,[(1-\alpha)p]^3-m^2\right. \\
& \cdot[1-(1-\alpha)p]^{2(m-1)}+(2m^2-2m-1)\,[1-(1-\alpha)p\,]^{2m} \\
& \left.+(2m-m^2-1)\,[1-(1-\alpha)p]^{2(m+1)}\right\}-\overline{\tau}^2_{HBH}(\alpha,p,m), \qquad (5.36)
\end{aligned}$$

where $\Theta(\Delta)$ is defined by Eq. (3.2) in Appendix R, and $\overline{\tau}_{SSP}(\alpha,p,m)$ and $\overline{\tau}_{HBH}(\alpha,p,m)$ are given by Eqs. (5.33) and (5.34), respectively.

Proof. The proof is provided in Appendix R. ■

Remarks on Theorem 5.4.1: Claim 1 derives formulas for multicast-tree bottleneck path distributions as a function of path length k, marginal link-marking probability p and dependency-degree factor α, and tree height m. Claim 2 examines the dynamic behavior of $\psi_d(P_k,\alpha,p,m)$ as p varies and observes that $\psi_d(P_k,\alpha,p,m)$ attains the unique maximum at p^* given by Eqs. (5.27) and (5.28), representing the link-marking probability that makes P_k the most likely multicast-tree bottleneck path. Claim 3 studies the behavior of

$\psi_d(P_k, \alpha, p, m)$ from the viewpoint of α and indicates that $\psi_d(P_k, \alpha, p, m)$ can be either monotonic or non-monotonic, depending on the given values of k and p. As long as k and p satisfy the conditions specified in Eq. (5.29), $\psi_d(P_k, \alpha, p, m)$ achieves the maximum at α^* given by Eq. (5.29).

Claim 4 reveals the fact that the Markov-chain dependency ($\alpha > 0$) reduces the probabilities for shorter paths to be the bottleneck while increasing the probabilities for longer paths to be the bottleneck. This probability shift is also shown to be balanced at the unique path, $P_{\widetilde{k}}$, where $\psi_d(P_{\widetilde{k}}, \alpha, p, m) \mid_{\alpha=0} = \psi_d(P_{\widetilde{k}}, \alpha, p, m) \mid_{\alpha>0}$, if the tree is high enough. This claim also derives the condition for the existence and uniqueness of $P_{\widetilde{k}}$ and the equation to compute the dependency-balanced path number $\widetilde{k}$ as a function of the given Markov chain dependency-factor α_0 and the link-marking probability p. Claim 5 and Claim 6 derives the closed-form expressions for the multicast-signaling delay means and variances for SSP and HBH as the functions of Δ, p, α, and m. In addition, Eqs. (5.26), (5.27), (5.28) (5.33), (5.34), (5.35), and (5.36) all reduce to the analytical results derived for the multicast signaling delay analysis under *independent* random-marking [45] by letting $\alpha = 0$, confirming the correctness of the dependence-degree modeling and these derived expressions in a sense.

5.5 Asymptotical Analysis of Link-Marking Markov Chains

Theorem 5.5.1 given below investigates the long-term behavior of the link-marking Markov chains based on the proposed Markov-chain dependency-degree model when m is large.

Theorem 5.5.1 *Consider the Markov chain $\{X_i\}$ defined by the link-marking states on both main- and branch-stream paths in the multicast tree specified by Definition 5.2.1. If* (i) *the dependency degree of $\{X_i\}$ is specified by the dependency-degree factor vector $\vec{\alpha} = (\alpha_1, \alpha'_1, \alpha_2, \alpha'_2, \alpha_3, \alpha'_3, \cdots)$ derived in Theorem 5.3.1,* (ii) *the link-marking probability vector is specified by $\vec{p} = (p_1, p'_1, p_2, p'_2, p_3, p'_3, \cdots)$ defined in Definition 5.2.1, and* (iii) *$\overline{p}$ and $\vec{\alpha}$ satisfy $0 < p_i = p'_i = p < 1$ and $0 \le \alpha_i = \alpha'_i = \alpha \le 1$, $\forall i$, respectively, such that $\{X_i\}$*

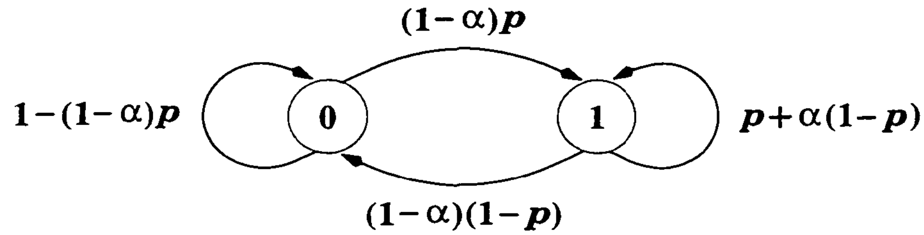


Figure 5.3: Markov chain model for dependent link-marking multicast flow control.

becomes a homogeneous Markov chain, then the following claims hold:

Claim 1. *The n-step transition probability matrix, denoted by $P^{(n)}$, of the homogeneous Markov chain $\{X_i\}$ is determined by:*

$$P^{(n)} \triangleq \left\{p_{jk}^{(n)}\right\} \triangleq \left\{\Pr\{X_{r+n} = k \mid X_r = j\}\right\} = \begin{bmatrix} 1-(1-\alpha^n)p & (1-\alpha^n)p \\ (1-\alpha^n)(1-p) & \alpha^n(1-p)+p \end{bmatrix} \quad (5.37)$$

where $j, k \in \{0,1\}$, $n \in \{0,1,2,\cdots\}$, $\forall r \geq 1$, $\Pr\{X_{r+n}=k \mid X_r=j\}|_{(r=i,n=1)}$ are given by Eqs. (5.19) through (5.22), and the Markov chain model for case of $P^{(n)}$ with $n = 1$ is shown in Figure 5.3;

Claim 2. *If $\alpha \in [0,1]$, then both link-marking states are ergodic, with*

$$\limsup_{n\to\infty} p_{jj}^{(n)} = \lim_{n\to\infty} p_{jj}^{(n)} > 0, \quad \lim_{n\to\infty} \sum_{r=1}^{n} p_{jj}^{(r)} = \infty, \quad (5.38)$$

where $j \in \{0,1\}$, and the recurring probability converges as follows:

$$\lim_{n\to\infty} p_{jj}^{(n)} = \begin{cases} \Pr\{X_k = j\} = 1-p, & \text{if } j = 0,\ \alpha \in [0,1); \\ \Pr\{X_k = j\} = p, & \text{if } j = 1,\ \alpha \in [0,1); \\ 1, & \text{if } j \in \{0,1\},\ \alpha = 1; \end{cases} \quad (5.39)$$

where $k \in \{1,2,\cdots\}$;

Claim 3. *If $\alpha \in [0,1)$, then the Markov chain $\{X_i\}$ is ergodic and its limiting probabilities exist and converge to the unique equilibrium state probabilities which are independent* *of both the initial state probabilities and dependency-degree α. The Markov chain's limiting probabilities, denoted by π_i, $i \in \{0, 1\}$, converge to the marginal link-marking probabilities as follows:*

$$\vec{\pi} \triangleq \begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} = \begin{bmatrix} (1-p) & p \end{bmatrix} \tag{5.40}$$

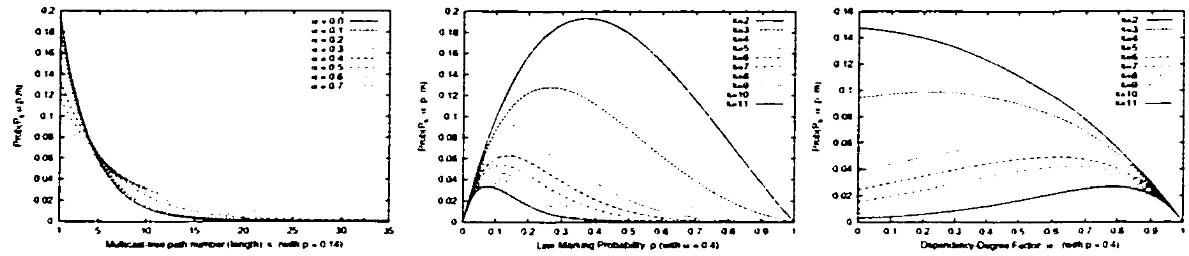
i.e., $\pi_0 = \Pr\{X_i = 0\} = (1-p)$ *and* $\pi_1 = \Pr\{X_i = 1\} = p$*;*

Claim 4. *If the Markov chain $\{X_i\}$ is perfectly dependent, i.e., $\alpha = 1$, then $\{X_i\}$ also converges to an equilibrium state, but the equilibrium state probabilities are not unique and are equal to the initial state probabilities. If the initial state probabilities are* $\Pr\{X_i = 0\} = 1 - p$ *and* $\Pr\{X_i = 1\} = p$*, then* $\pi_0 = 1 - p$ *and* $\pi_1 = p$ *still hold.*

Proof. The proof is provided in Appendix W. ■

Remarks on Theorem 5.5.1: Claim 1 fully specifies the long-term behavior of the Markov chain and determines the distribution of a bottleneck path in the homogeneous case. Claim 2 classifies the link-marking states as the dependency-factor α varies. It also shows that the Markov-chain state recurring probabilities converge asymptotically to the marginal link-marking probabilities (see Eq. (5.39)), if the Markov chain is not perfectly dependent ($\alpha \neq 1$).

Claim 3 ensures that the Markov-chain dependency-degree modeling converges asymptotically, and the long-term behavior of the resulting Markov chain is stable. Also, the ergodicity of the Markov chain enables us to evaluate its various statistics (ensemble average) through the sample averages in simulations or implementations. Moreover, this claim shows that the limiting probabilities converge to the marginal link-marking probabilities $\Pr\{X_i = x_i\}$, where $x_i \in \{0, 1\}$. This is also expected, because π_0 and π_1 represent the long-term proportion of the Markov chain remaining at state 0 and 1, respectively, and is consistent with the definitions of $\Pr\{X_i = 0\}$ and $\Pr\{X_i = 1\}$, which verifies the validity of the Markov-chain dependency-degree model. Claim 4 says that when $\alpha = 1$, i.e., the link-marking state is perfectly dependent, the equilibrium state distribution still exists,



(a) $\psi_d(P_k, \alpha, p, m)$ vs. k (b) $\psi_d(P_k, \alpha, p, m)$ vs. p (c) $\psi_d(P_k, \alpha, p, m)$ vs. α

Figure 5.4: Impact of path length k, link-marking probability p, and dependency-degree α on bottleneck path probability distribution $\psi_d(P_k, \alpha, p, m)$.

but is not unique, depending on the initial state probabilities. This is expected because when $\alpha = 1$, the Markov chain $\{X_i\}$ has two isolated classes (see Figure 5.3). So, it is not irreducible, and thus is no longer ergodic.

5.6 Numerical and Simulation Evaluations

Based on the analytical results derived thus far, various multicast signaling delay properties are evaluated numerically as described as follows.

5.6.1 Multicast-Tree Bottleneck Path Distribution $\psi_d(P_k, \alpha, p, m)$

Figure 5.4(a) plots $\psi_d(P_k, \alpha, p, m)$ against path length k while varying the Markov-chain dependency-degree factor α. $\psi_d(P_k, \alpha, p, m)$ is found to be a strictly monotonic decreasing function of k for both the independent ($\alpha = 0$) and dependent ($\alpha > 0$) cases. This is expected because the longer the bottleneck path, the more likely it will be dominated by shorter paths, as described in Definition 5.2.2.

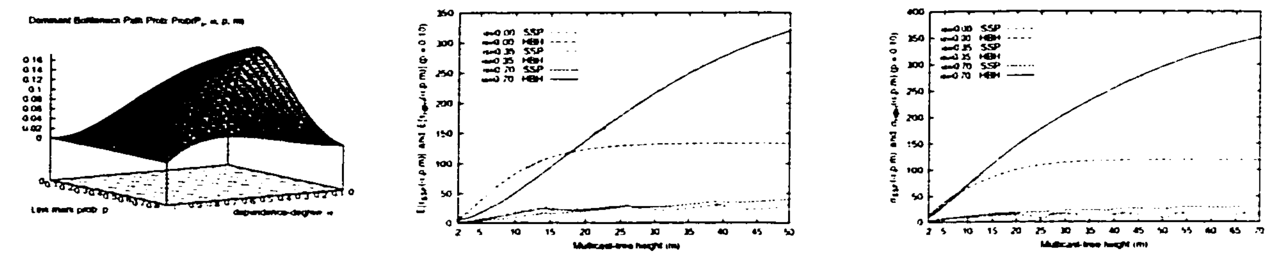
Compared to the independent-marking case, the marking dependency is found to reduce the probability for shorter paths (with $k \leq 4$) to be the bottleneck path while increasing the probability for longer paths (with $k > 5$). This verifies Claim 4 of Theorem 5.4.1, and the dependency-balanced path number: $\tilde{k}$ is found to be around 4 and 5. Figure 5.4(a) also shows that the larger α, the more this probability shifts from shorter paths to longer ones.

This is because the stronger the link-marking dependency, the larger the probability that all links stay in the *same* congestion state. This trend is also shown in Figure 5.6(a), plotting $\psi_d(P_k, \alpha, p, m)\,|_{\alpha=0} - \psi_d(P_k, \alpha, p, m)\,|_{\alpha=\alpha_0>0}$ against k for different values of $\alpha = \alpha_0$.

We now analyze why the bottleneck probability shifts from shorter paths to longer ones as α increases, as shown in Figure 5.4(a). Theorems 5.2.1 and 5.4.1 state that for P_k to be the multicast bottleneck, all links on shorter paths $P_{k'}$ ($k' < k$) must be un-congested and P_k's last two links L_k or L'_{k+1} must be congested. Thus, $\psi_d(P_k,\alpha,p,m)$ is contributed by two events, $\{X_k = 1 \cup X'_{k+1} = 1\}$ and $\{\bigcap_{i=1}^{k-1}(X_i = 0, X'_{i+1} = 0)\}$, which must occur at the same time. But, the link-marking dependency reduces the probability contribution from $\{X_k = 1 \cup X'_{k+1} = 1\}$ while increasing that from $\{\bigcap_{i=1}^{k-1}(X_i = 0, X'_{i+1} = 0)\}$. Then for $\alpha > 0$ the decaying rate of $\psi_d(P_k, \alpha, p, m)$ as k increases is slower than that for the case of $\alpha = 0$. Compared to the case of $\alpha = 0$, when k is small ($k \leq 4$), the decrease of probability contribution from $\{X_k = 1 \cup X'_{k+1} = 1\}$ due to $\alpha > 0$ cannot be compensated for by the increase in that from $\{\bigcap_{i=1}^{k-1}(X_i = 0, X'_{i+1} = 0)\}$. So, $\psi_d(P_k, \alpha, p, m)\,|_{\alpha>0} < \psi_d(P_k, \alpha, p, m)\,|_{\alpha=0}$ for small k ($k \leq 4$). When k is large ($k > 5$), the gain in probability contribution from $\{\bigcap_{i=1}^{k-1}(X_i = 0, X'_{i+1} = 0)\}$ is larger than the drop in that from $\{X_k = 1 \cup X'_{k+1} = 1\}$ due to $\alpha > 0$. Thus $\psi_d(P_k, \alpha, p, m)\,|_{\alpha>0} > \psi_d(P_k, \alpha, p, m)\,|_{\alpha=0}$ for a large k ($k > 5$). When k becomes very large, both $\psi_d(P_k, \alpha, p, m)\,|_{\alpha>0}$ and $\psi_d(P_k, \alpha, p, m)\,|_{\alpha=0}$ converges to zero. So, $\psi_d(P_k, \alpha, p, m)\,|_{\alpha>0} = \psi_d(P_k,\alpha, p, m)\,|_{\alpha=0}$ as $k \to \infty$, which is confirmed by Figure 5.4(a).

But, no matter how $\psi_d(P_k, \alpha, p, m)$ shifts as α changes, the normalization condition given by Eq. (5.24) is always satisfied; this is verified by the fact that the area under each plot for any given α always sums to 1 as shown in Figure 5.4(a).

Figure 5.4(b) shows that $\psi_d(P_k, \alpha, p, m)$ is inversely proportional to path length k, also verifying the above observations. Figure 5.4(b) also shows that there exists a unique maximum $\psi^*_d(P_k, \alpha, p^*, m)$ for any given k, verifying Claim 2 of Theorem 5.4.1. Figure 5.4(c) indicates that for any given α, the larger the path length k, the smaller $\psi_d(P_k, \alpha, p, m)$.



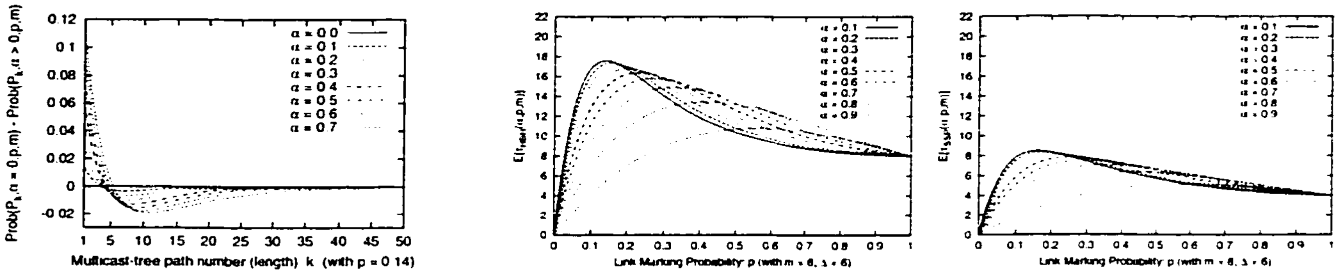
(a) $\psi_d(P_k, \alpha, p, m)$ vs. (α, p) (b) $\overline{\tau}_{SSP}(\alpha, p, m)$, $\overline{\tau}_{HBH}(\alpha, p, m)$ vs. m (c) $\sigma_{SSP}(\alpha, p, m)$, $\sigma_{HBH}(\alpha, p, m)$ vs. m

Figure 5.5: Impact of dependency-degree factor α, link-marking probability p, and multicast-tree height m on bottleneck path probability $\psi_d(P_k, \alpha, p, m)$ and bottleneck RM-cell RTT means and standard deviations.

Figure 5.4(c) also indicates that $\psi_d(P_k, \alpha, p, m)$ is not always a monotonic function of α, but there can be a unique maximum $\psi_d^*(P_k, \alpha^*, p, m)$ as long as the given path length k and p satisfy the conditions in Eq. (5.29). As k gets larger, α^* increases. These also validate Claim 3 of Theorem 5.4.1. Figure 5.5(a) shows a more complete dynamic-behavior picture of $\psi_d(P_k, \alpha, p, m)$ as a function of two independent variables (α, p). Figure 5.5(a) shows that $\psi_d(P_k, \alpha, p, m)$ always has the maximum along the p-axis direction as α varies from 0 to 1. In contrast, $\psi_d(P_k, \alpha, p, m)$ can have the maximum along the α-axis direction only for a certain range of p values which satisfy the conditions given in Eq. (5.29) in Theorem 5.4.1 for a given k.

5.6.2 Delay Statistics for HBH and SSP Schemes under the Dependent Markings

Figure 5.5(b) plots the means, $\overline{\tau}_{SSP}(\alpha, p, m)$ and $\overline{\tau}_{HBH}(\alpha, p, m)$ calculated by Eqs. (5.33) and (5.34), respectively, against m for different α's. We observe that $\overline{\tau}_{HBH}(\alpha, p, m)$ is much larger, and increases much faster, than $\overline{\tau}_{SSP}(\alpha, p, m)$ as shown in Figure 5.5(b). Moreover, $\overline{\tau}_{HBH}(\alpha, p, m)$ is more sensitive to α than $\overline{\tau}_{SSP}(\alpha, p, m)$. Figure 5.5(b) also shows that, as compared to the HBH's average RTT $\overline{\tau}_{HBH}(\alpha, p, m)$, the SSP's average RTT $\overline{\tau}_{SSP}(\alpha, p, m)$

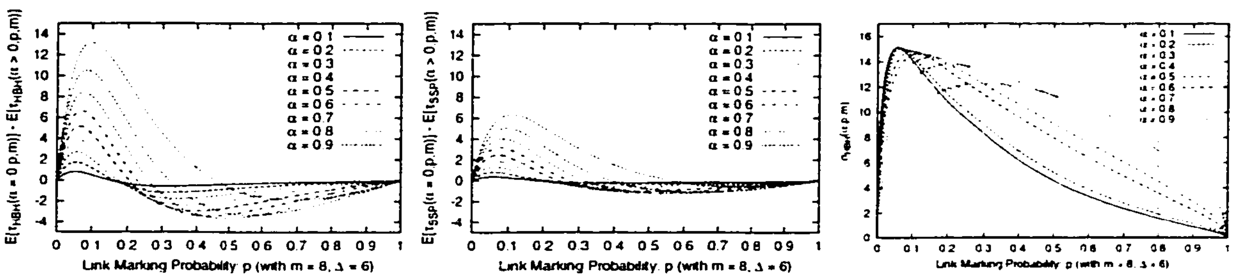


(a) $\psi_d(P_k,\alpha,p,m)\,|_{\alpha=0} - \psi_d(P_k,\alpha,p,m)\,|_{\alpha>0}$ vs. p (b) $\overline{\tau}_{HBH}(\alpha,p,m)$ vs. p (c) $\overline{\tau}_{SSP}(\alpha,p,m)$ vs. p

Figure 5.6: Impact of dependency-degree factor α and link-marking probability p on bottleneck path probability $\psi_d(P_k,\alpha,p,m)$ shift and bottleneck RM-cell RTT means.

is virtually independent of m and α. Figure 5.5(b) also shows that for longer paths ($m > 20$), the larger α, the larger the means while for shorter paths ($m < 12$), the larger α, the smaller the means. These verify that the bottleneck path probabilities shift from shorter to longer paths as α increases, which is shown in Figure 5.6(a).

In Figure 5.5(c), the standard deviations, $\sigma_{SSP}(\alpha,p,m)$ and $\sigma_{HBH}(\alpha,p,m)$ given by Eqs. (5.35) and (5.36), respectively, are plotted against m while varying α. As shown in Figure 5.5(c), $\sigma_{HBH}(\alpha,p,m)$ is found to be much larger, and increase much faster, than $\sigma_{SSP}(\alpha,p,m)$ as m increases. Again, $\sigma_{HBH}(\alpha,p,m)$ is much more sensitive to α than $\sigma_{SSP}(\alpha,p,m)$. Thus, the bottleneck RM-cell RTT for SSP scales much better than that for HBH with respect to the multicast-tree height and structure. Figure 5.5(c) also shows that SSP's multicast RTT variation $\sigma_{SSP}(\alpha,p,m)$ is virtually independent of both m and α, as compared to HBH's RTT variation $\sigma_{HBH}(\alpha,p,m)$. Figure 5.5(c) also shows that for longer paths ($m \geq 10$), the larger α, the larger the variances while for shorter paths ($m < 8$), the larger α, the smaller the variances, also verifying that the bottleneck probabilities shift from shorter to longer paths as α increases, which is also shown in Figure 5.6(a).



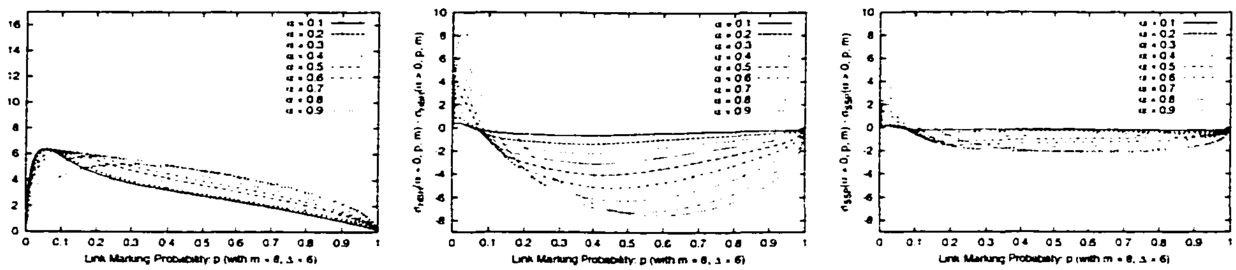
(a) $\varepsilon_m^{HBH}(\alpha, p, m)$ vs. p (b) $\varepsilon_m^{SSP}(\alpha, p, m)$ vs. p (c) $\sigma_{HBH}(\alpha, p, m)$ vs. p

Figure 5.7: Impact of dependency-degree factor α and link-marking probability p on approximation error under independent markings assumption and bottleneck RM-cell RTT standard deviations.

5.6.3 Impact of Link-Marking Dependency Degree (α) on Multicast Signaling Delays

Figures 5.6(b) and (c) plot the means of multicast signaling delays $\overline{\tau}_{HBH}(\alpha, p, m)$ and $\overline{\tau}_{SSP}(\alpha, p, m)$, respectively, against the network traffic load p, while varying the Markov-chain dependency-degree factor α. We have the following observations: (1) there is a unique maximum for each of $\overline{\tau}_{HBH}(\alpha, p, m)$ and $\overline{\tau}_{SSP}(\alpha, p, m)$ with respect to p, which is consistent with the unique maximum of $\psi_d(P_k, \alpha, p, m)$ in Claim 2 of Theorem 5.4.1; (2) the maximizers for $\overline{\tau}_{HBH}(\alpha, p, m)$ and $\overline{\tau}_{SSP}(\alpha, p, m)$ shift from a small to large value of p as α increases; (3) also, as α increases, $\overline{\tau}_{HBH}(\alpha, p, m)$ and $\overline{\tau}_{SSP}(\alpha, p, m)$ become less sensitive to the network traffic load p; (4) $\overline{\tau}_{HBH}(\alpha, p, m)$ is about two times larger than $\overline{\tau}_{SSP}(\alpha, p, m)$ for all values of p calculated under our parameter settings.

To evaluate the approximation error of the multicast signaling delay analysis where the "independent marking" is assumed while the actual congestion markings are not independent, Figures 5.7(a) and (b) plot the approximation errors in terms of means between the multicast-signaling delay analyses under the dependent ($\alpha > 0$) and independent ($\alpha = 0$) markings. The approximation errors in terms of the means of multicast signaling delay are



(a) $\sigma_{SSP}(\alpha, p, m)$ vs. p (b) $\varepsilon_d^{HBH}(\alpha, p, m)$ vs. p (c) $\varepsilon_d^{SSP}(\alpha, p, m)$ vs. p

Figure 5.8: Impact of dependency-degree factor α and link-marking probability p on bottleneck RM-cell RTT standard deviations and the approximation error under independent markings assumption.

---

defined as follows:

$$\varepsilon_m^{HBH}(\alpha, p, m) \triangleq \overline{\tau}_{HBH}(\alpha, p, m)\mid_{\alpha=0} - \overline{\tau}_{HBH}(\alpha, p, m)\mid_{\alpha>0}; \tag{5.41}$$

$$\varepsilon_m^{SSP}(\alpha, p, m) \triangleq \overline{\tau}_{SSP}(\alpha, p, m)\mid_{\alpha=0} - \overline{\tau}_{SSP}(\alpha, p, m)\mid_{\alpha>0}. \tag{5.42}$$

We have the following observations: (1) the maxima of both $\varepsilon_m^{HBH}(\alpha, p, m)$ and $\varepsilon_m^{SSP}(\alpha, p, m)$ are monotonically increasing functions of α, implying that the approximation error increases as the dependency degree increases; (2) both $\varepsilon_m^{HBH}(\alpha, p, m)$ and $\varepsilon_m^{SSP}(\alpha, p, m)$ are not monotonic functions of traffic load p, but change from a positive to a negative value as p increases from 0 to 1, indicating that the analysis under the independent assumption over-estimates the mean delay for small p, while under-estimating the mean delay for large p; (3) the approximation error for HBH is more than two times higher than that for SSP. In general, the above analyses show that the approximation error in terms of multicast signaling delay mean resulting from the independent assumption is not negligible, and thus justifies the necessity of a Markov-chain-based marking-dependency analysis.

Similar observations on the impact of dependent markings can be drawn for the multicast signaling delay variations as shown in Figures 5.7(c) and 5.8(a) (for delay variations) and Figures 5.8(b) and (c) (for the approximation error of delay variations defined by Eqs. (5.43) and (5.44)) under HBH and SSP, respectively. The approximation errors (see Figures 5.8(b) and (c)) in terms of the standard deviations of multicast signaling delay are defined as follows:

$$\varepsilon_d^{HBH}(\alpha, p, m) \triangleq \sigma_{HBH}(\alpha, p, m)\,|_{\alpha=0} - \sigma_{HBH}(\alpha, p, m)\,|_{\alpha>0}; \tag{5.43}$$

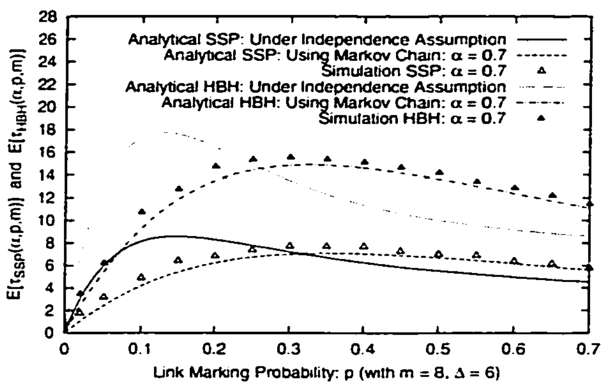
$$\varepsilon_d^{SSP}(\alpha, p, m) \triangleq \sigma_{SSP}(\alpha, p, m)\,|_{\alpha=0} - \sigma_{SSP}(\alpha, p, m)\,|_{\alpha>0}\,. \tag{5.44}$$

5.6.4 Simulation Results

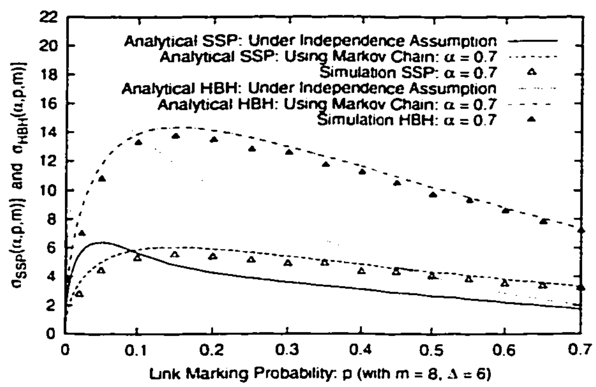
To confirm and evaluate the accuracy of the proposed Markov-chain and Markov-chain dependency-degree models, using the NetSim event-driven simulator [42] we also simulated a network with concurrent multiple multicast/unicast VCs (Virtual Circuits) and multiple bottlenecks. Figures 5.9(a) and (b) plot the multicast signaling delay mean and standard deviation over the traffic load p, respectively, drawn from both simulations and analytical results. Figures 5.9(a) and (b) also show that the simulation results agree well with the analytical results (for the case of $\alpha = 0.7 > 0$) which use the proposed Markov-chain and Markov-chain dependency degree models, thus verifying the accuracy of modeling and analytical results derived based on the Markov chain and Markov-chain dependency degree models (in Section 5.2 through Section 5.4). In contrast, the simulation results disagree with the analytical results obtained under the independent-marking assumption as indicated by the the large approximation error as shown in Figures 5.9(a) and (b). So, the simulation experiments also quantitatively justify the necessity of the Markov-chain- and Mark-chain dependency-degree-models-based multicast-signaling delay analysis.

5.7 Conclusion

In this chapter, we proposed the dependent statistical modeling approaches to analyze the performance of a class of multicast feedback-synchronization signaling algorithms. Specifically, we developed a Markov-chain model to characterize the multicast signaling de-



(a) $\overline{\tau}_{SSP}(\alpha, p, m)$ and $\overline{\tau}_{HBH}(\alpha, p, m)$ vs. p



(b) $\sigma_{SSP}(\alpha, p, m)$ and $\sigma_{HBH}(\alpha, p, m)$ vs. p

Figure 5.9: Comparison of the simulated delay means and standard deviations with the analytical results.

lay when the congestion markings of different links are dependent. Using this model, we derived a set of general expressions for calculating the probability distributions of individual paths in a multicast tree being the multicast-tree bottleneck. The derived Markov chain is shown to be able to reach an equilibrium, and its limiting state distributions converge to the marginal link-marking probabilities when the Markov chain is irreducible.

We also developed a Markov-chain dependency-degree model to quantify and evaluate the dependency between different link congestion markings. Using the proposed Markov-chain dependency-degree model, we derived a set of equations to compute all one-step transition probabilities as functions of the marginal link-marking probabilities and the Markov-chain dependency-degree factors. The proposed Markov-chain and Markov-chain dependency-degree models are generic and thus can be used not only for signaling delay analysis, but for other Markov-chain-based analyses extracted from other applications as well.

Using these two models we derived the first and second moments of a multicast-signaling delay for both HBH and SSP flow control signaling schemes, respectively, when link-markings are dependent. The obtained numerical evaluations also showed that the dependency degree factor tends to shift the bottleneck from a shorter path to a longer one, which is consistent with the definition of the positive link-marking dependency imposed by the nature of multicast flow control signaling. The analytical results have also been confirmed by the simulation experiments.

CHAPTER 6

OPTIMIZATION-BASED MULTICAST FLOW CONTROL USING VIRTUAL M-ARY FEEDBACK

6.1 Motivation and Overview of the Proposed Scheme

6.1.1 Motivation

Multicast has a wide spectrum of applications, such as software distribution, multimedia conferencing, and distance learning/collaboration [46–55]. Like in unicast, flow control also plays a very important role in multicast service over the best-effort networks [56–68], such as the Internet. Most flow-control schemes are typically structured as a closed-loop feedback control system [69, 70], where the traffic source adjusts its transmission rate or window size based upon the congestion feedback generated by the receivers and routers in the network. Congestion feedback can be in different forms, such as packet drops in TCP, ECN-bits in RED-gateways, DEC-bits in DECbit-Network, and ER-field or CI-bit in ATM ABR. We categorize the feedback into two types: (1) binary feedback where the traffic source decides on every flow-control action using only a single bit feedback, like TCP's packet drops, and duplicate ACKs, RED's ECN-bit, ABR's CI-bit, etc., and (2) M-ary feedback where each flow-control decision at the source is derived from multiple-bit feedback, like ABR's Explicit-Rate (ER) feedback in ATM networks. While binary feedback minimizes the flow-control signaling overhead, it has the drawback of low dynamic stability and low bandwidth-utilization efficiency, because it only implements coarse-grained flow control. For instance, TCP halves its (window) (sending rate) as long as it "sees" indication of packet-drop/loss, regardless whether this congestion is caused by a short-term, less severe congestion, or by a long-term severer congestion. In contrast, M-ary feedback can offer much higher flow-control performance because it applies fine-grained flow-control, accurately adapting the source rate to the bandwidth available at the bottleneck. However, M-ary feedback-based flow control is more expensive in both router complexity and bandwidth consumption for flow-control signaling. This problem becomes even severer in case of multicast because multicast usually incurs much higher volume of flow-control feedback signaling traffic, particularly when the number of multicast-tree branches is large.

Moreover, multicast also introduces several other new challenges that were not encountered in unicast. First, simultaneous congestion feedbacks from all receivers can cause *feedback implosion* [13] at the source and branch routers, especially when the multicast tree is large. Hence, it is important to consolidate the congestion feedbacks at each branch point, and only the consolidated result is sent upstream. Second, since feedbacks via different downstream paths may arrive at the branch point at significantly different times, the feedback consolidation must be synchronized at the branch point before sending the consolidated result upstream to avoid the feedback noise problem [20]. Third, the flow-control scheme must be able to keep track of the *most-congested path* which changes dynamically and dictates the source rate to ensure that every receiver receives the same data. Finally, the multicast feedback consolidation or fusion mechanism must be able to derive the *congestion level* of the most-congested path such that the source can perform fine-grained flow control. This is crucial for multicast over a wide-area network with a large round-trip time (RTT), because either inaccurate or coarse-grain feedback information can significantly degrade the stability and responsiveness of multicast flow control, bandwidth utilization, and the overall flow-control performance.

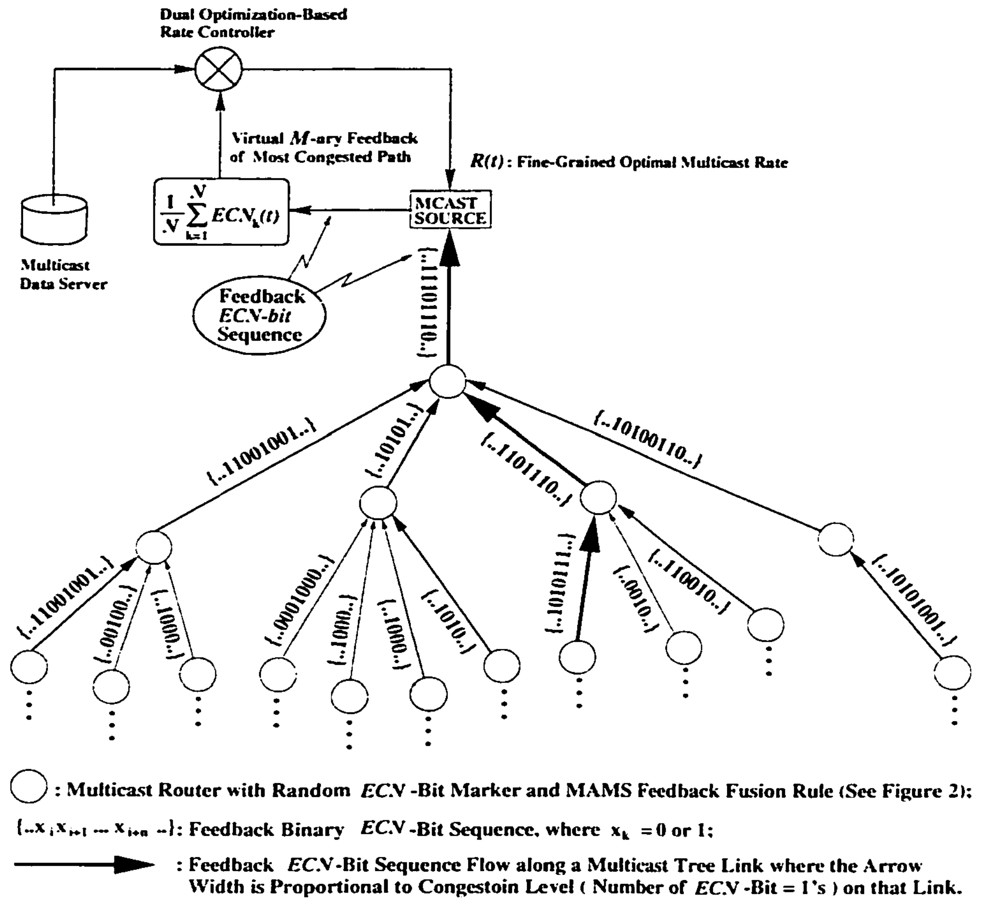


Figure 6.1: The architecture of the proposed scheme.

6.1.2 Overview of the Proposed Scheme

To solve the above-mentioned problems of multicast flow control, while retaining the benefits of both the binary and M-ary feedback flow-control schemes without their demerits, we propose an optimization-based multicast flow-control scheme using *virtual M-ary* (VMARY) feedback that performs as well as an explicit M-ary feedback mechanism while only employing binary feedback. Figure 6.1 illustrates the proposed scheme which consists of the four major components: (1) dual optimization-based rate controller at source, (2) multicast-tree marking probability generator at source, (3) feedback ECN-bit random marker at routers, and (4) feedback ECN-bit feedback sequence fusion rule at branch routers. Each of these components is described below.

6.1.2.1 Dual Optimization-Based Multicast Rate Controller

Conceptually, the flow-control problem can be formulated as a distributed optimization problem [71,72], as in the recent work for unicast [35,37,38,40,43,44] and for layered multirate multicast [73,74]. The objective of the optimization-based flow control is to maximize the global utility in terms of bandwidth utilization subject to link-bandwidth constraints. The previous research on optimization-based flow control focused only on either unicast [35,37,38,40,43,44], or layered multirate multicast [73,74]. By contrast, we will focus on single-rate multicast flow control which is much simpler [75] and more cost-effective in implementation and resource management than the layered multi-rate multicast flow control, because the latter requires multiple multicast groups to implement a single multicast session, which is not resource-efficient for source, receivers, and routers [76]. Moreover, the single-rate multicast flow control meets the requirements of some important applications, such as reliable content distribution, and can scale well with the number of receivers in the multicast group [75,77]. As a result, the single-rate multicast flow control has received considerable attention [75,77].

As analyzed in Section 6.2, although the optimization-based multicast flow-control problem is analytically tractable, it requires direct coordination among all the source-rate controllers associated with multiple concurrent multicast sessions. This implies that the optimization-based multicast flow control is a *coupled* problem, making it practically unsolvable. Therefore, as was done in [71,72], we apply the duality principle to decompose the multicast flow-control problem into a number of *independent* subproblems, which can then be solved independently and in parallel. However, the important extensions/differences distinguishing our approach from others' are: (1) our approach deals with multicast flow control, instead of unicast flow control over a single fixed path; (2) our dual optimization is based on single-rate multicast flow control; (3) our flow-control model performs the dual optimization only over the multicast-tree's most-congested paths, labeled with the thick arrowed lines in Figure 6.1, i.e., a subset of links dominate the optimization decisions at all multicast sources. We show that the dual optimization problem defined over the multicast-tree's most-congested paths have the same optimization solution and objective-function value as those for the original optimization problem defined on the entire multicast tree. This significantly reduces the computational complexity in solving the multicast flow-control problem.

6.1.2.2 Derivation of VMARY Feedback on the Most-Congested Path

The second component of our multicast flow-control scheme is how to derive the bandwidth constraints along the most-congested path in a multicast tree to make the optimization decision at the source. This requires information on the level of congestion of the multicast tree that needs to be derived from the consolidated feedback from the receivers and routers. We use a congestion detection and collection mechanism at all multicast routers based on feedback ECN-bit random marking. Especially, we develop a multicast-tree marking probability generator ($\frac{1}{N}\sum_{k=1}^{N} ECN_k(t)$) at each source as shown in Figure 6.1, to calculate the path marking probability which is proportional to the sum of average queue lengths along the most-congested path in a multicast tree. We call this feedback the *virtual* M-ary (VMARY) feedback, as it only uses a sequence of N consecutive feedback random marks — binary ECN-bits — to extract the fine-grained congestion-level information $\frac{1}{N}\sum_{k=1}^{N} ECN_k(t)$, which is usually coded and conveyed as an M-ary ($\log_2 M$-bit, $M > 2$) packet to represent the sum of average queue lengths along the most-congested path. Such M-ary feedback enables the fine-grained optimization-based multicast flow control at the source, but is expensive to implement. Using a history of ECN-bits to derive a control decision is a generalization of the 2-bit ECN sequence-based α-control [6] as well as the DEC-bit scheme [78], but our ECN is based on random (instead of deterministic) marking. REM (Random Early Marking) [35,37,38,40,43,44] applies a similar approach, but has been used only in unicast. The proposed multicast-tree marking probability generator is detailed in Section 6.3.1.

6.1.2.3 VMARY Feedback Signaling Protocol

Multicast flow-control signaling coordinates the optimization-decision makers at different sources of multiple concurrent multicast sessions and optimization-constraint detectors/collectors at routers while performing the optimization in a distributed manner. Multicast flow-control signaling plays a crucial role in the *distributed* optimization flow control since it measures and conveys the congestion information from the location where the optimization constraint occurs, to the location where the optimization constraint comes into play in making the optimization decision at the source. To deal with the new signaling problems imposed by multicast, we develop a multicast signaling feedback mechanism which is composed of the following two major elements: (1) feedback ECN-bit random marker that measures the congestion level at each link/router in the multicast tree, and (2) feedback ECN-bit sequence fusion rule which consolidates, synchronizes, and dynamically tracks the most-congested feedback ECN-bit flows from all downstream paths at each branch router.

Feedback ECN-bit Random Marker at Each Router: This is necessary to implement the proposed VMARY feedback. Note that congestion-level measure at each link based on random ECN-bit marking transforms the queue-length (congestion-level) information (i.e., the marking probability) into a sequence of coded ECN-bits, which is then transformed back to the queue-length (congestion-level) information, or the marking probability, from the feedback ECN-bit sequence received at the source. The ECN-bit random marker works similarly to RED/REM, where it marks the feedback ACK-packet with a probability proportional to the average queue length at that output link. On the other hand, the proposed ECN-bit random marker differs from RED/REM in that it only marks the ECN-bit in *feedback ACK-packets* corresponding to every $K \geq 1$[1] *forward data packets* received by each receiver — unlike RED/REM which marks ECN-bit in each forward data packet passed through each router. This is because the marked ECN-bit in a forward data packet at an

[1] K is used to represent the tradeoff between the congestion level and signaling traffic volume.

upstream branch router may be redundantly duplicated multiple times as the data packet passes through downstream branch routers, even when the downstream branches/receivers are not congested at all. Consequently, when all of these redundant ECN-bits are returned by all receivers, the resultant congestion signal will over-throttle the source rate unnecessarily, a phenomenon similar to the multiple-path problem [79]. In contrast, the ECN-bits carried by the feedback ACK-packet in the proposed signaling protocol can precisely inform the multicast source of the congestion level on each path in the multicast tree. Section 6.3.1 details the proposed feedback ECN-bit random marker.

Feedback ECN-bit Sequence Fusion Rule at Routers: This element needs to perform three main functions: (1) consolidate feedback ECN-bit sequences, (2) synchronize the feedback consolidation, and (3) identify the most-congested path/branch in the multicast tree. We achieve these by developing a Maximum-MArk-Select (MAMS) fusion rule which is implemented at each branch router as shown in Figure 6.1 (also in Figure 6.2 for more details). It generates one consolidated ECN-bit if and only if it receives at least one feedback ECN-bit from each of connected downstream branches, such that the feedback consolidation is soft synchronized (similar to [6]) even when the feedbacks from different receivers arrive at the branch point at significantly different times. The consolidated ECN-bit is selected from the path whose last N consecutive feedback ECN-bits contain the maximum number of 1's, ensuring the selection of ECN-bit sequence of the most congested path because the percentage of marks in the last N consecutive feedback ECN-bits reflects the average-queue length or congestion level on each path. Note that it is essential to dynamically identify the most-congested path/branch at each router for implementation of the proposed single-rate optimization-based flow control at the source since the single-rate reliable multicast flow control must guarantee the receiver on the most-congested path to receive the same data copy as all other receivers in the multicast tree. The proposed optimal feedback ECN-bit sequence fusion rule and the implementation issues are described in Section 6.3.1.

6.1.3 Chapter Organization

The rest of the chapter is organized as follows. Section 6.2 models multicast flow-control control scheme and justifies the rational of the proposed control model. Section 6.3 describes the implementation of the proposed scheme and the optimal fusion rule at branch routers. Section 6.4 investigates the effect of fusion register on the multicast flow-control performance. Section 6.5 presents the numerical analysis of the proposed feedback fusion mechanism, while Section 6.6 conducts the simulation evaluation, confirming analytical results and observations. The chapter concludes with Section 6.7.

6.2 The Optimization Model of Multicast Flow Control

6.2.1 The System Model

To formulate the multicast flow as an optimization control problem and justify the rational behind the proposed scheme, we establish the flow-control system model by first introducing the following definitions.

Definition 6.2.1 *A multicast network of finite bandwidth shared by multiple elastic (best-effort) traffic sources under the multicast flow control, satisfies the following conditions.*

C1. *All links of the multicast network are unidirectional and indexed by a set of* $\mathcal{L} \triangleq \{1, 2, \cdots, L\}$ *with link* ℓ*'s bandwidth capacity* $\mu_\ell \in (0, +\infty)$, $\forall \ell \in \mathcal{L}$;

C2. *All elastic traffic sources are persistent (using as much bandwidth as available) and indexed by a set of* $\mathcal{S} \triangleq \{1, 2, \cdots, S\}$ *where the transmission rate of source* s *is denoted by* $r_s \in I_s \triangleq [m_s, M_s]$, $0 < m_s < M_s < \infty$, *and* $\mathcal{I} \triangleq \{I_s \mid s \in \mathcal{S}\}$, $\forall s \in \mathcal{S}$;

C3. *A multicast-connection tree with source* s *at the root, denoted by* $MT(s)$, *is characterized by a 7-tuple* $(\mathcal{L}(s), \mathcal{L}_k(s), \mathcal{L}_{k^*}(s), \mathcal{L}^*, m_s, M_s, n)$ *where* $n \geq 1$ *is the number of branches of the multicast source* s *(*$n = 1$ *represents a unicast, a special case of multicast), and* $\mathcal{L}(s) \subseteq \mathcal{L}$ *are links of* $MT(s)$, $s \in \mathcal{S}$, $\mathcal{L}_k(s)$, $k \in \{1, 2, \cdots, n\}$, *is the link subset constituting the k-th path from s to its k-th receiver such that* $\mathcal{L}(s) = \bigcup_{k\in\{1,\cdots,n\}} \mathcal{L}_k(s)$*;* $\mathcal{L}_{k^*}(s)$ *is the link subset constituting the* **most congested path** *among the n paths from s to receivers, which is defined by Definition 6.2.3;*

C4. *For each link* $\ell \in \mathcal{L}$*, define* $\mathcal{S}(\ell) \triangleq \{s \in \mathcal{S} \mid \ell \in \mathcal{L}(s)\}$ *as the set of sources that use link* ℓ*, such that* $\ell \in \mathcal{L}(s)$ *if and only if* $s \in \mathcal{S}(\ell)$*.* ■

Remarks on Definition 6.2.1: Condition **C1** specifies a finite-bandwidth multicast network while **C2** characterizes the traffic transported through the multicast network. **C3** defines the descriptors and structure of a multicast-tree. **C4** describes the relations between multicast source and the corresponding links in $MT(s)$.

Applying the nonlinear programming approaches [71, 72, 80, 81] to the multicast network model in Definition 6.2.1, we can formulate the multicast flow control as an optimization problem as follows.

Definition 6.2.2 *The primal optimization problem, denoted by* **P***, of multicast flow control for source* $s \in \mathcal{S}$ *over the multicast network defined by Definition 6.2.1 is to choose source rate* r_s*,* $\forall s \in \mathcal{S}$*, such that*

$$\mathbf{P}: \quad \max_{r_s\in I_s, s\in\mathcal{S}} \sum_{s\in\mathcal{S}} U_s(r_s) \tag{6.1}$$

$$\text{subject to} \sum_{s\in\mathcal{S}(\ell)} r_s \leq \mu_\ell, \quad \forall \ell \in \mathcal{L} \triangleq \{1, 2, \cdots, L\} \tag{6.2}$$

where $U_s(r_s) : \Re_+ \longmapsto \Re$ *is a utility function for source s; source s is said to attain a utility* $U_s(r_s)$ *when it transmits at rate* $r_s \in I_s$*; and* $U_s(r_s)$ *is chosen to be increasing and strictly concave in its argument in the feasible solution set.* ■

Remarks on Definition 6.2.2: The constraint given by Eq. (6.2) says that the total source rate at any link ℓ is less than its capacity μ_ℓ. A unique maximizer, called the *primal* *optimal solution*, exists since the objective function is chosen to be strictly concave and hence is continuous, and the feasible solution set is compact.

6.2.2 Multicast-Tree Bottleneck Path

In unicast flow control, the source rate is regulated by the feedback from the most congested link/router which has the minimum *available* bandwidth along the path from source to destination. A natural extension of this strategy to multicast flow control is to adjust the source rate to the minimum available bandwidth share of the multicast-tree's most congested path that the traffic source has sensed from the feedback. This is the key feature for data applications that require lossless transmission. To explicitly model these features for the multicast flow control, we introduce the following definition.

Definition 6.2.3 *The* ***multicast-tree bottleneck path*** *(also simply called* ***multicast-tree bottleneck****) is the* **most congested path** *whose congestion feedback at the source dictates (or dominates) the source rate-control decisions. If letting $\mathcal{L}_{k^*}(s)$ be the subset of links which constitutes the* **most congested path** *among the n paths from source s to receivers for any given $MT(s)$, $\forall s \in \mathcal{S}$, then the most congested path's index determined by*

$$k^* = \arg \min_{k\in\{1,\cdots,n\},\, \ell\in\mathcal{L}_k(s)} \left\{ \mu_\ell - \sum_{s\in\mathcal{S}(\ell)} r_s \right\}. \tag{6.3}$$

Thus, $\mathcal{L}^* \triangleq \bigcup_{s\in\mathcal{S}} \mathcal{L}_{k^*}(s) = \{1, 2, \cdots, L^*\}$ represents the subset of links each of which is part of at least one of the most congested path in S multicast trees, where $L^* \triangleq \|\mathcal{L}^*\|$ is the the cardinality of $\mathcal{L}^*$.

Remarks on Definition 6.2.3: The most-congested-path-dominant multicast flow-control policy is widely used in the single-rate data multicast flow control [75–77,82], such as *representative acker* in *pgmcc* [77] and *current limiting receiver* (CLR) in [75]. Making use of this feature, we can simplify the data multicast flow control by adapting the source rate only to the most congested path in a multicast tree, which is formally justified by the following theorem.

Theorem 6.2.1 *If the multicast flow control for source $s \in \mathcal{S}$ over the multicast network defined by Definition 6.2.1 is formulated as the primal-optimization problem* $\mathbf{P}$ *specified by Definition 6.2.2, then* $\mathbf{P}$*'s primal-optimization solution and optimal objective-function value are the same, respectively, as those of another primal-optimization problem,* $\mathbf{P}^*$*, which is defined as follows.*

The multicast flow control for source $s \in \mathcal{S}$ over the multicast network defined by Definition 6.2.1 is to choose source rate r_s, $\forall s \in \mathcal{S}$, such that

$$\mathbf{P}^* : \quad \max_{r_s \in I_s, s \in \mathcal{S}} \sum_{s \in \mathcal{S}} U_s(r_s) \tag{6.4}$$

$$\text{subject to} \sum_{s \in \mathcal{S}(\ell)} r_s \leq \mu_\ell \quad \forall \ell \in \mathcal{L}_* \triangleq \bigcup_{s \in \mathcal{S}} \mathcal{L}_{k^*}(s) = \{1, 2, \cdots, L^*\}, \tag{6.5}$$

Proof. The proof is given in Appendix X. ∎

Remarks on Theorem 6.2.1: This theorem enables us to solve the multicast flow-control problem by solving $\mathbf{P}^*$, instead of $\mathbf{P}$, where $\mathbf{P}$ is much more computationally complex and more difficult to implement, than $\mathbf{P}^*$. We will henceforth only focus on solving $\mathbf{P}^*$.

6.2.3 A Separable Optimization Structure for Multicast Flow Control

As shown in Theorem 6.2.1, if constraints $\sum_{s \in \mathcal{S}(\ell)} r_s \leq \mu_\ell$, $\forall \ell \in \mathcal{L}_* \triangleq \bigcup_{s \in \mathcal{S}} \mathcal{L}_{k^*}(s) = \{1, 2, \cdots, L^*\}$ given by Eq. (6.5) were not present, it would be possible to decompose $\mathbf{P}^*$ into S independent subproblems as follows:

$$\max_{r_s \in I_s, s \in \mathcal{S}} \sum_{s \in \mathcal{S}} U_s(r_s) = \sum_{s \in \mathcal{S}} \max_{r_s \in I_s} U_s(r_s). \tag{6.6}$$

However, solving $\mathbf{P}^*$ subject to the constraint Eq. (6.5) requires direct coordination between multiple source-rate controllers, which makes $\mathbf{P}^*$ a *coupled* optimization problem. As a result, directly solving $\mathbf{P}^*$ is not practical, and hence, we propose the optimal multicast-rate control by applying the Duality Theory [80, 81], which solves $\mathbf{P}^*$'s dual-optimization problem $\mathbf{D}^*$ that can decouple the rate-control coordination among multicast traffic sources.

The theorem given below proves the feasibility of separating the multicast flow control into S independent subproblems, and derives a distributed optimization algorithm which can implement the multicast dual-optimization rate control in a parallel manner.

Theorem 6.2.2 *If the multicast flow control for source $s \in \mathcal{S}$ over the multicast network defined by Definition 6.2.1 is achieved by the primal optimization model specified by Definition 6.2.2, then the following claims hold.*

Claim 1. *$\mathbf{P}^*$'s dual optimization problem is determined by*

$$\mathbf{D}^*: \quad \min_{\lambda_\ell \geq 0,\, \ell \in \mathcal{L}^*} D^*(\vec{\lambda}_*) \tag{6.7}$$

where the objective function $D^*(\vec{\lambda}_*)$ is defined by a Lagrangian function $L(\vec{r}, \vec{\lambda}_*)$

$$D^*(\vec{\lambda}_*) \triangleq \max_{r_s \in I_s, s \in \mathcal{S}} L(\vec{r}, \vec{\lambda}_*) \triangleq \sum_{s \in \mathcal{S}} B_s^*(\lambda_{k^*}^s) + \sum_{\ell \in \mathcal{L}^*} \lambda_\ell \mu_\ell \tag{6.8}$$

where

$$B_s^*(\lambda_{k^*}^s) = \max_{r_s \in I_s, s \in \mathcal{S}} \{U_s(r_s) - r_s \lambda_{k^*}^s\}, \quad \forall s \in \mathcal{S} = \{1, 2, \cdots, S\} \tag{6.9}$$

$$\lambda_{k^*}^s = \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell = \max_{k \in \{1, \cdots, n\}} \{\lambda_k^s\} = \max_{k \in \{1, \cdots, n\}} \sum_{\ell \in \mathcal{L}_k(s)} \lambda_\ell. \tag{6.10}$$

Claim 2. *The optimal solution to $\mathbf{D}^*$ exists, is unique, and equals the optimal solution to $\mathbf{P}^*$;*

Claim 3. *The dual-optimization solution to $\mathbf{D}^*$ for multicast flow control can be solved by a distributed gradient projection algorithm, which yields the following iterative equation where for each multicast source $s \in \mathcal{S}$, the Lagrange multiplier λ_ℓ at time $(t+1)$ for each link $\ell \in \mathcal{L}_{k^*}(s)$ is determined by*

$$\lambda_\ell(t+1) = \left[\lambda_\ell(t) + \gamma \left(\sum_{s \in \mathcal{S}(\ell),\, \ell \in \mathcal{L}_{k^*}(s)} r_s\left(\vec{\lambda}_*(t)\right) - \mu_\ell\right)\right]^+, \forall \ell \in \mathcal{L}_{k^*}(s), \forall s \in \mathcal{S} \tag{6.11}$$

where $\gamma > 0$ is the step size of the distributed gradient projection algorithm, and $[Z]^+ \triangleq \max\{Z, 0\}$ is a projection function.

Claim 4. $\mathbf{D}^*$ *decomposes* $\mathbf{P}^*$ *into* S *independent subproblems in terms of the aggregate utility and the aggregate constraints.*

Proof. The proof is given in Appendix Y. ■

Remarks on Theorem 6.2.2: Claim 1 formulates the multicast flow-control problem under the general Lagrange Dual Theory, and gives a concrete and workable analytical model to achieve the optimal multicast flow control. The first term of the dual objective function $D(p)$ is decomposed into S separable subproblems Eqs. (6.9) and (6.10). Claim 2 is the direct application of Duality Theorem: (1) if the primal-optimization problem $\mathbf{P}^*$ has an optimal solution, the dual-optimization problem $\mathbf{D}^*$ also has an optimal solution and the two optimal values are same; (2) the multicast rate vector $\vec{r}_o \in \mathcal{I}$ are the primal-optimal solutions for $\mathbf{P}^*$ and Lagrange multiplier vector $\vec{\lambda}^o_*$ are the dual-optimal solution for $\mathbf{D}^*$ if and only if

$$\max_{r_s \in I_s, s \in \mathcal{S}} L(\vec{r}, \vec{\lambda}^o_*) = L(\vec{r}_o, \vec{\lambda}^o_*) = \min_{\lambda_\ell \geq 0, \ell \in \mathcal{L}_*} L(\vec{r}_o, \vec{\lambda}_*). \tag{6.12}$$

The existence and uniqueness of the dual-optimal solution is guaranteed by the fact that the objective function of $\mathbf{P}^*$ is strictly concave. Claim 3 derives an iterative algorithm which can be implemented in a distributed fashion to calculate the congestion-level (the bandwidth-constraint information) at all routers in the multicast network. This algorithm gives a standard solution for the non-constrained optimization problem transformed from a nonlinear-constrained optimization problem by Lagrange-Multiplier and Lagrange-Duality theorems. It shows that the dual-optimization problem also decomposes the multicast optimization flow-control problem in terms of constraints because each link's Lagrange multiplier is iteratively calculated based upon the previous value of this link's Lagrange multiplier, and is also independent of all other links' Lagrange multipliers. Claim 4 is crucially important because it decomposes the coupled optimal optimization problem into separate subproblems, making it possible to implement $\mathbf{D}^*$ by a distributed algorithm in parallel. In particular, given the minimizer vector of Lagrange multiplier $\vec{\lambda}^o_*$ (obtained by solving Eq. (6.7)), each individual multicast traffic source can solve Eq. (6.9) for optimal multicast flow-control rates $\vec{r}_o$ independently or separately without the need to coordinate with other sources. The correlation among the multiple multicast flow-controllers, due to sharing of the same multicast network, is captured by the Lagrange multiplier vector $\vec{\lambda}_*$, which serves as a coordination signal to align individual optimalities defined by Eq. (6.9) with the global optimality described by Eq. (6.4).

6.3 Virtual M-ary Feedback Signaling and Multicast Flow Control

Theorem 6.2.2 transforms the multicast flow control over the multicast network defined by Definition 6.2.1 into a distributed computing system. It treats the multicast source $s \in \mathcal{S}$ and all links $\ell \in \mathcal{L}_{k^*}(s)$ on its most congested path as multiple processors connected by the most congested path to solve the dual-optimization problem $\mathbf{D}^*$. In each iteration, each multicast source $s \in \mathcal{S}$ individually solves Eq. (6.9) and communicates the thus-obtained result $r_s(\vec{\lambda}_*)$ to links $\ell \in \mathcal{L}_{k^*}(s)$ on the most congested path in $MT(s)$. Links $\ell \in \mathcal{L}_{k^*}(s)$ then update their Lagrange multipliers λ_ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$, using Eq. (6.11), and communicate the new λ_ℓ back to the source s, and the procedure repeats. The rate-control algorithms specified by Eqs. (6.9), (6.10), and (6.11) only provide the first component of a multicast flow-control scheme (the second component is the flow-control signaling). We now describe how the multicast source s and network links $\ell \in \mathcal{L}_{k^*}(s)$ on the most congested path of $MT(s)$ communicate through the multicast flow-control signaling protocol that we propose below.

6.3.1 The Virtual M-ary Feedback Multicast Signaling Protocol

The multicast flow control formalized by Eq. (6.7) is an M-ary feedback-based multicast flow control, and thus requires an M-ary feedback signaling protocol where both the multicast source rate $r_s(\vec{\lambda}_*)$ and Lagrange multipliers λ_ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$, must be expressed and transmitted as multiple-bit signaling messages between the multicast source s and all links $\ell \in \mathcal{L}_{k^*}(s)$ on the most congested path of $MT(s)$. A straightforward solution to multicast signaling is to periodically use the control packets to *explicitly* exchange the multicast rate $r_s(\vec{\lambda}_*)$ and Lagrange multipliers λ_ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$ between the multicast source s and the links in the most congested path. This approach is conceptually simple, but complex and expensive in implementation for the following reasons. First, it is very time-/space-consuming for each router to compute the aggregate arrival rates at each output link, which are required to compute $\lambda_\ell(t+1)$ as specified by Eq. (6.11). Second, it is also very time-/space-consuming for each router to identify the most congested branch at its input port, which is specified by Eq. (6.3) to determine the most congested path from the multicast source s, among all output-links from the branch. Third, the computations of Lagrange multiplier λ_ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$ at the input ports of branch routers on the most congested path is also complicated and expensive in time and space. Finally, explicitly sending $r_s(\vec{\lambda}_*)$ and feeding back λ_ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$ will create a large volume of multicast signaling traffic, incurring significant bandwidth overhead. Thus, this solution is impractical to use for the multicast flow control formulated as the dual-optimization problem $\mathbf{D}^*$.

To overcome this difficulty, we propose a *virtual M-ary feedback signaling protocol*, which only uses binary feedback, but can implement the M-ary feedback-based multicast flow control of $\mathbf{D}^*$ defined by Eq. (6.7). The proposed virtual M-ary feedback multicast signaling protocol consists of three components: (1) feedback ECN-bit link ACK-random-marker at each multicast-branch output link, (2) optimal ECN-bit-sequence fusion at each multicast-branch input link port, and (3) multicast-tree marking probability generator at the multicast source.

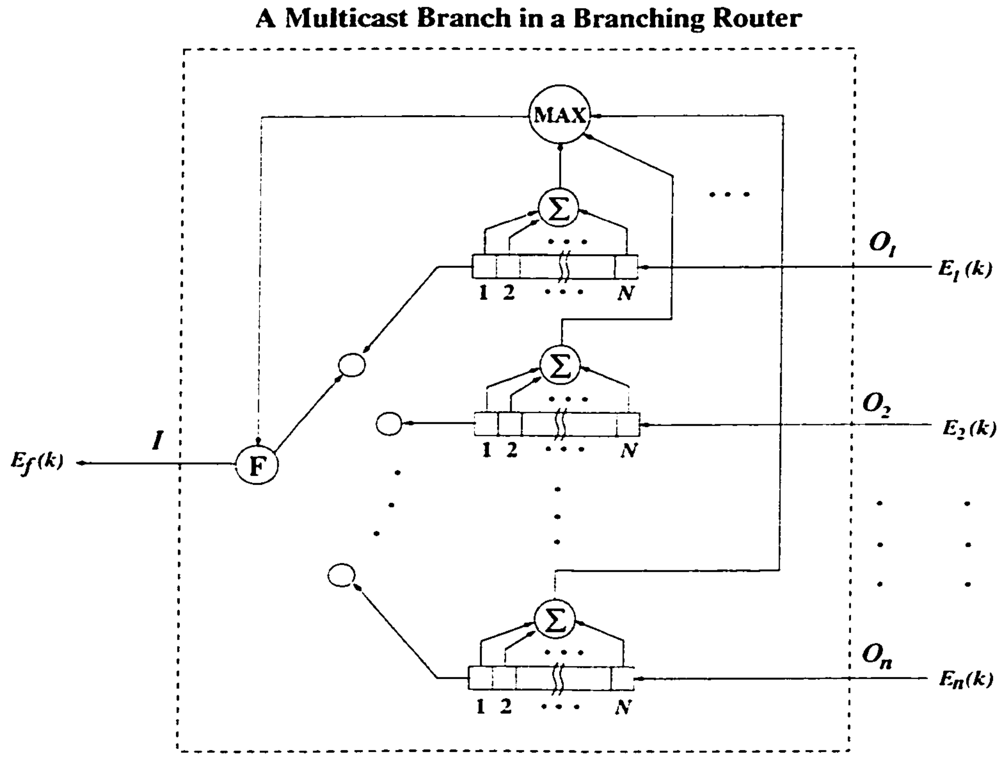


Figure 6.2: The MAx-Mark-Select (MAMS) fusion rule for consolidating feedback ECN sequence $\{E_i(k)\}$'s.

6.3.1.1 ECN-Bit ACK-Random-Marker at Each Multicast-Branch Output Port

This works in a way similar to REM or RED-based flow control in terms of computation of marking probabilities. Each output port sets its marking probability — exponentially as in REM or linearly as in RED — proportional to the average queue length at that output link. Using the average queue length, instead of the instantaneous queue length, preserves the advantage of allowing transient bursts in the router. However, there is a big difference that distinguishes the virtual M-ary feedback multicast signaling protocol from REM and RED: it doesn't mark forward data packets as in REM or RED, but each router in the multicast tree randomly marks only the feedback ACK-packet as it passes through the branch output port toward the receiver that generated this ACK-packet. Each receiver in the multicast tree, $MT(s)$, acknowledges receipt of each forward data packet by creating and sending toward the source s an ACK-packet which contains an ECN-bit. The reason of marking each feedback ACK-packet, instead of a forward data packet, is because if ECN-bit of a data packet is marked by an upstream branch router, this ECN-bit will be redundantly replicated multiple times as this data packet traverses downstream branch-routers, even when some downstream branches are not congested at all. As a result, when all these redundant ECN-bits are returned by all receivers, the resultant congestion signal will over-throttle the source rate unnecessarily. In contrast, ECN-bits generated and carried by the feedback ACK-packet in the proposed multicast signaling protocol can accurately signal the multicast source s the level of congestion along each path in $MT(s)$.

6.3.1.2 Optimal ECN-Bit-Sequence Fusion at Each Multicast-Branch Input Port

We design an optimal ECN-bit-sequence fusion mechanism at each multicast-branch input-link port, which is connected to multiple branch output link ports, as shown in Figure 6.2. The three main purposes of ECN-bit-sequence fusion are to:

(1) consolidate feedback ECN-bits to avoid the feedback implosion and hence scale well with multicast-tree size;

(2) perform the synchronization of feedback ACKs from all branch-paths to avoid feedback noise;

(3) identify the most congested path in the multicast tree to implement the dual-optimization multicast flow control as specified by Theorem 6.2.2.

The ECN-bit-sequence fusion mechanism at each branch input port is composed of n ECN-bit-sequence fusion shift registers of length N bits each, corresponding to the n branches of this input port. The ECN-bit-sequence fusion shift registers in Figure 6.2 can be easily implemented in either software or hardware since it only needs simple operations. Each fusion shift register contains/maintains and shifts, from-right-to-left, N ECN-bits consecutively received from the feedback ACK-packets of that branch. When a feedback ACK-packet arrives at the output port O_i, $i = 1, \cdots, n$ from the i-the downstream branch and if it is not marked (i.e., no congestion indication), then it is marked randomly with the marking probability proportional to the average queue length in that output port; an already-marked ECN-bit is kept unchanged. The "processed" ECN-bit is then put into the rightmost bit of the i-th register after left-shifting the entire register by one bit. To identify the most congested path in a given multicast tree $MT(s)$, we design a MAximum Mark Selected (MAMS) fusion rule to consolidate the feedback ECN-bit sequence, as shown in Figure 6.2. After each shift-register receives at least one feedback ACK-packet from all connected downstream branches where the soft-synchronization [2] is achieved, the leftmost bit of the shift register which has the maximum number of 1's among all shift registers in this input port is selected and put into the consolidated feedback ACK-packet generated by this input port. Then, this newly-generated consolidated feedback ACK-packet is forwarded to its upstream router. Consequently, the proposed ECN-bit-sequence fusion mechanism generates a consolidated ACK-packet *if and only if* all of its downstream branches receive at least one feedback ACK-packet, and the output feedback ECN-bit sequence $\{E_f(k)\}$ at the input port I will follow the pattern of $\{E_i(k)\}$ at the output port O_i which contains the maximum number of 1's in the last N consecutive feedback ACK-packets, which corresponds to the most congested branch path from this branch point.

6.3.1.3 Multicast-Tree Marking Probability Generator at the Multicast Source

It has been shown in [71, 72] that the Lagrange multipliers λ_ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$ can be used as a link congestion level indicator, which is proportional to the output link's average queue length $\overline{q}_\ell(t)$:

$$\lambda_\ell(t) = \gamma \overline{q}_\ell(t) \tag{6.13}$$

This is also verified by Theorem 6.2.2, where if the aggregate arrival rates, or equivalently the rate-mismatch between the aggregate arrival rates and the bottleneck bandwidth, are too

```
On receipt each feedback ACK-packet:
    calculate the new multicast-tree marking probability;
        if (left_most_bit = 1) and (ACK_ECN = 0);
            mcast_tree_mark_prob := mcast_tree_mark_prob − 1/N;
        elseif (left_most_bit = 0) and (ACK_ECN = 1);
            mcast_tree_mark_prob := mcast_tree_mark_prob + 1/N;
        endif;
```

Figure 6.3: Pseudocode for the multicast marking probability calculation algorithm.

large, then the Lagrange multipliers $\lambda_\ell(t+1)$, $\forall \ell \in \mathcal{L}_{k^*}(s)$ derived from Eq. (6.11) increase. Then, the increased Lagrange multipliers $\lambda_\ell(t+1)$, $\forall \ell \in \mathcal{L}_{k^*}(s)$ will require reduction of the optimal multicast-source rates r_s, $\forall s \in \mathcal{S}$, as shown in Eq. (6.9). On the other hand, if the sending rates are over-reduced, then Eq. (6.11) generates smaller Lagrange multipliers $\lambda_\ell(t+1)$, $\forall \ell \in \mathcal{L}_{k^*}(s)$, which then lead to larger optimal sending rates r_s, $\forall s \in \mathcal{S}$ determined by Eq. (6.9). Based on this observation, we can use the average queue length $\overline{q}_\ell(t)$ at link ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$ to derive λ_ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$ for all links on the most congested path in $MT(s)$, $s \in \mathcal{S}$.

According to the REM-based ECN-bit-link ACK-random-marker described in Section 6.3.1, the ECN-bit marking probability at time t at link ℓ, denoted by $p_\ell(t)$, is exponentially to the average queue length $\overline{q}_\ell(t)$;

$$p_\ell(t) = 1 - \phi^{-\gamma \overline{q}_\ell(t)}, \qquad \forall \ell \in \mathcal{L}_{k^*}(s) \tag{6.14}$$

where ϕ is a constant. When feedback ACK-packets of multicast source $s \in \mathcal{S}$ pass through the output link ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$, they are *independently* marked with the probability $p_\ell(t)$ defined in Eq. (6.14). Thus, when a feedback ACK-packet arrives at the multicast source s at time t, the probability $p_*^s(t)$ that its ECN-bit is marked is exponential to the *sum* of average queue-lengths of all output links ℓ, $\forall \ell \in \mathcal{L}_{k^*}(s)$, along the most congested path of

$MT(s)$ which is given by:

$$p_*^s(t) = 1 - \prod_{\ell \in \mathcal{L}_{k^*}(s)} (1 - p_\ell(t)) = 1 - \phi^{-\sum_{\ell \in \mathcal{L}_{k^*}(s)} \gamma \bar{q}_\ell(t)}. \tag{6.15}$$

We define $p_*^s(t)$ as the *multicast-tree marking probability* for $MT(s)$ since it measures the sum of average queue-lengths at all links along the most congested path from source s, and thus dominates the congestion control decision at $s \in \mathcal{S}$.

Plugging Eq. (6.13) into Eq. (6.15), we obtain

$$\lambda_{k^*}^s(t) = \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell(t) = -\log\left(1 - p_*^s(t)\right). \tag{6.16}$$

Eq. (6.16) gives a useful formula to derive the sum of Lagrange multipliers specified by Eq. (6.10), which is required to solve for the optimal multicast rate r_s specified by Eq. (6.9), from the multicast-tree marking probability $p_*^s(t)$. Since $p_*^s(t)$ is the marking probability for the feedback ACK-packet returning via the most congested path, it can be estimated from a sequence of N consecutive ECN-bits, $\{E_k(t)\}$ at time t (note that here the symbols used to represent feedback ECN-bit sequence for $\{E_k(t)\}$ are different from those used in Figure 6.2 where k is the time index and i is the branch index while here k is the time or number index of the k-th ECN-bit of interest within an N-bit long "multicast-tree marking probability calculation window"[2] at the multicast source and t is the time when the multicast-tree marking probability $\hat{p}_*^s(t)$ is calculated), generated by the optimal ECN-bit-sequence fusion mechanism described in Section 6.3.1 as follows:

$$\hat{p}_*^s(t) = \frac{1}{N}\sum_{k=1}^{N} E_k(t). \tag{6.17}$$

The pseudo-code for implementing the multicast-tree marking probability generator specified by Eq. (6.17) at the multicast source is given in Figure 6.3. Letting $\hat{p}_*^s(t) = p_*^s(t)$ and plugging Eq. (6.17) into Eq (6.16) yield the final formula to compute the Lagrange multipliers specified by Eq. (6.10) to solve for the optimal multicast rate r_s specified by Eq. (6.9)

[2] The multicast-tree marking probability generator can also be implemented by using an ECN-bit sequence shift register.

```
On receipt each feedback ACK-packet:
    calculate the new optimal rate;
        if mcast_tree_mark_prob = 0;
            r_s := max_rate;
        elsif mcast_tree_mark_prob = 1;
            r_s := min_rate;
        else mcast_tree_mark_prob := 1;
            λ_ℓ := − log(1 − mcast_tree_mark_prob) / log φ;
            r_s := max{min{w_s/λ_ℓ, max_rate}, min_rate};
        endif mcast_tree_mark_prob := 0;
```

Figure 6.4: Pseudocode for the optimal multicast rate control algorithm.

from the N-bit long ECN-bit sequence sent back via feedback ACK-packets as follows:

$$\lambda^{s}_{k^*}(t) = \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell(t) = -\frac{\log\left(1 - \frac{1}{N}\sum_{k=1}^{N} E_k(t)\right)}{\log \phi}. \tag{6.18}$$

The pseudo-code for the multicast source rate-control algorithm based on Eq (6.18) is summarized in Figure 6.4.

6.4 Length of the Optimal Feedback Fusion Register

For the proposed multicast feedback fusion, the length N of ECN-bit shift-register is a critical design parameter. This design problem is also associated with, but has not yet been addressed in, the unicast optimization-based flow control [35,37,38,40,43,44]. However, this problem gets much more complicated for multicast feedback fusion. Clearly, neither too large nor too small a value of N is desired. Too small an N can lower bandwidth-utilization efficiency because the multiple multicast ECN-bit sequences stored in too short an ECN-bit register can generate an excessive number of 1's in the aggregated/consolidated ECN-bit sequence at the output end (left-most bit of the shift-register) which will eventually over-reduce the sending rate, lowering bandwidth utilization. On the other hand, a too large an N can filter out the rapid, short-term variation of traffic congestion, thus lowering the responsiveness/adaptiveness of the feedback fusion mechanism to the change of traffic pattern, which can either lower network utilization, or make networks over-congested due to lack of responsiveness to the increased congestion condition.

The above observations indicates the existence of an optimal ECN-bit shift register size, denoted by N^*, which makes an optimal tradeoff of the above two opposing effects. However, the design of optimal N^* turns out to be a non-trivial problem, because it is involved not only with the feedback fusion rule and multicast-tree topology/size, but also with the network traffic dynamics and congestion levels/burstiness, which are typically random and not predictable *a priori*. To quantitatively and accurately capture these trade-offs in selecting optimal buffer length N^*, we introduce the following definition.

Definition 6.4.1 *Under the proposed multicast feedback fusion rule, the achieved multicast bandwidth efficiency, denoted by a random variable F_η, is characterized by*

$$F_\eta \triangleq 1 - \frac{1}{N} \max_{1 \le j \le n} \sum_{i=1}^{N} X_{ij}\, e^{-\frac{1}{n}\left\{N - \max_{1 \le j \le n} \sum_{i=1}^{N} X_{ij}\right\}}, \tag{6.19}$$

and the achieved adaptiveness of the proposed multicast feedback fusion, denoted by a random variable F_α, is characterized by

$$F_\alpha \triangleq e^{-\frac{1}{n}\left\{N - \max_{1 \le j \le n} \sum_{i=1}^{N} X_{ij}\right\}}, \tag{6.20}$$

and the achieved multicast fusion utility function, denoted by a random variable F_μ, is characterized by

$$F_\mu \triangleq F_\eta + F_\alpha - 1 = \left(1 - \frac{1}{N} \max_{1 \le j \le n} \sum_{i=1}^{N} X_{ij}\right) e^{-\frac{1}{n}\left\{N - \max_{1 \le j \le n} \sum_{i=1}^{N} X_{ij}\right\}}, \tag{6.21}$$

where $X_{ij} \in \{0, 1\}$ is the i-th ECN bit in the j-th branch filter of the feedback ECN filter array. ■

Remarks on Definition 6.4.1: The above definition defines the metrics by which one can measure and derive the optimal ECN-bit shift-register length N^*.

Based on Definition 6.4.1, the following theorem derives a formula to calculate the statistical metrics in deriving N^* for any multicast-tree topology and link-marking probability.

Theorem 6.4.1 *Consider an ECN-filter array with buffer size equal to N and the branching fan-out factor equal to n. If random markings at different branches are independent, then the means and variances of F_η, F_α, and F_μ are determined by the following expressions, respectively:*

$$\begin{cases} E\left[F_\eta\right] = \displaystyle\sum_{i=0}^{N-1} \left\{ \left\{ 1 - \frac{i}{N} e^{-\frac{1}{n}[N-i]} \right\} \right. \\ \qquad \left. \cdot \displaystyle\sum_{\{(y_1, y_2, \cdots, y_n) \,|\, \max_{1 \le j \le n}\{y_j\} = i\}} \prod_{j=1}^{n} \binom{N}{y_j} p_j^{y_j} \left(1 - p_j\right)^{N - y_j} \right\}, \\ Var\left[F_\eta\right] = \displaystyle\sum_{i=0}^{N-1} \left\{ \left\{ 1 - \frac{i}{N} e^{-\frac{1}{n}[N-i]} \right\}^2 \right. \\ \qquad \left. \cdot \displaystyle\sum_{\{(y_1, y_2, \cdots, y_n) \,|\, \max_{1 \le j \le n}\{y_j\} = i\}} \prod_{j=1}^{n} \binom{N}{y_j} p_j^{y_j} \left(1 - p_j\right)^{N - y_j} \right\} - E^2\left[F_\eta\right], \end{cases} \tag{6.22}$$

and

$$\begin{cases} E\left[F_\alpha\right] = \displaystyle\sum_{i=0}^{N-1} \left\{ \left\{ e^{-\frac{1}{n}[N-i]} \right\} \right. \\ \qquad \left. \cdot \displaystyle\sum_{\{(y_1, y_2, \cdots, y_n) \,|\, \max_{1 \le j \le n}\{y_j\} = i\}} \prod_{j=1}^{n} \binom{N}{y_j} p_j^{y_j} \left(1 - p_j\right)^{N - y_j} \right\}, \\ Var\left[F_\alpha\right] = \displaystyle\sum_{i=0}^{N-1} \left\{ \left\{ e^{-\frac{2}{n}[N-i]} \right\} \right. \\ \qquad \left. \cdot \displaystyle\sum_{\{(y_1, y_2, \cdots, y_n) \,|\, \max_{1 \le j \le n}\{y_j\} = i\}} \prod_{j=1}^{n} \binom{N}{y_j} p_j^{y_j} \left(1 - p_j\right)^{N - y_j} \right\} - E^2\left[F_\alpha\right], \end{cases} \tag{6.23}$$

and

$$\begin{cases} E\left[F_\mu\right] = \sum_{i=0}^{N-1}\left\{\left\{\frac{N-i}{N}e^{-\frac{1}{n}[N-i]}\right\} \cdot \sum_{\{(y_1,y_2,\cdots,y_n)\,|\,\max_{1\le j\le n}\{y_j\}=i\}} \prod_{j=1}^{n}\binom{N}{y_j}p_j^{y_j}\left(1-p_j\right)^{N-y_j}\right\}, \\ Var\left[F_\mu\right] = \sum_{i=0}^{N-1}\left\{\left\{\left(\frac{N-i}{N}\right)^2 e^{-\frac{2}{n}[N-i]}\right\} \cdot \sum_{\{(y_1,y_2,\cdots,y_n)\,|\,\max_{1\le j\le n}\{y_j\}=i\}} \prod_{j=1}^{n}\binom{N}{y_j}p_j^{y_j}\left(1-p_j\right)^{N-y_j}\right\} - E^2\left[F_\mu\right], \end{cases} \tag{6.24}$$

where $p_j = \Pr\{X_{ij} = 1\}$ *and* $y_j \in \{0, 1, 2, \cdots, N\}$ *for* $j = 1, 2, \cdots, n$, *and* $X_{ij} \in \{0, 1\}$ *is the* i*-th ECN bit in the* j*-th feedback ECN-bit register.*

Proof. See Appendix Z. ■

Remarks on Theorem 6.4.1: This theorem provides a set of closed-form expressions for the first and second moments of bandwidth efficiency and adaptiveness of the proposed feedback fusion rule. These equations are useful because they can help network designers compute the optimal ECN buffer length N^* for different link marking probabilities and multicast-tree topologies. The assumption that random markings at different branches are independent is reasonable, because the ECN-bit is only randomly marked in feedback ACK packets, which traverse different branch paths from different multicast receivers, and arrive at the multicast source independently. N equals the marking-probability computation window size. The marking probability p_j differs for different multicast branches, but we assume that $p_{ij} = p_j$ along each multicast branch is constant within the probability computation window N because the ECN-bit sequence in the window N is used to compute/estimate the single marking probability p for that window.

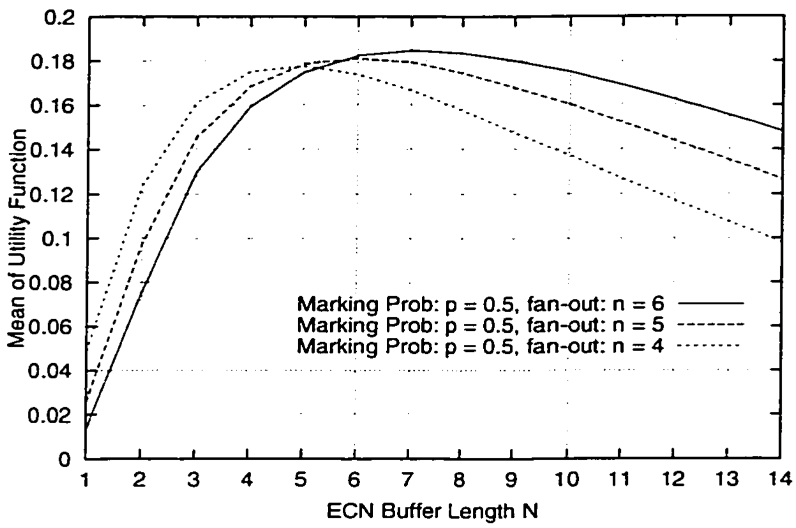


Figure 6.5: Mean utility function $E[F_\mu]$ vs. ECN buffer size N with different fan-out factors n.

6.5 Numerical Evaluation for the Feedback Fusion Rule

Figure 6.5 plots the mean utility function $E[F_\mu]$ against the ECN shift-register length N for different multicast fan-out factors n. We observe that $E[F_\mu]$ is not a monotonic function of N, but there exists a unique maximum for $E[F_\mu]$ given any fan-out factor n. This was expected because either too large or too small a value of N can lower the bandwidth-utilization efficiency. Figure 6.5 also indicates that the optimal ECN shift-register size N^*'s for fan-out factor $n = 4, 5, 6$ are $5, 6, 7$, respectively, showing that the larger the fan-out factor, the larger N^*. This is reasonable because a large number of branches will need a longer ECN shift-register length to achieve higher bandwidth-utilization efficiency.

Figure 6.6 plots the mean of utility function $E[F_\mu]$ against ECN-bit shift-register length N for different link marking probabilities p. We also observe that $E[F_\mu]$ is not a monotonic function of N, and there exists a unique maximum for $E[F_\mu]$ given any fan-out factor n. Figure 6.6 also indicates that the optimal ECN shift-register length N^* for link marking probabilities $p = 0.4, 0.5, 0.6$ are $4, 5, 7$, respectively, showing that the larger the link marking probability p, the larger N^*. This is reasonable because a large link marking probability

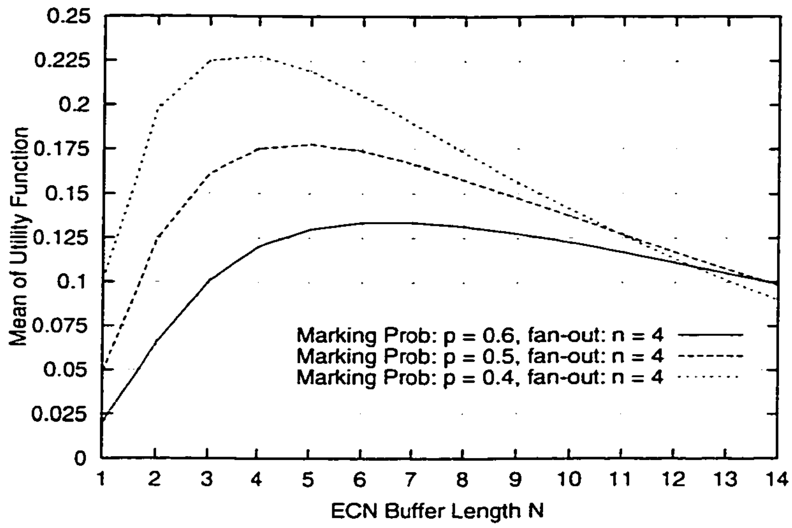


Figure 6.6: Mean of utility function $E[F_\mu]$ vs. ECN buffer size N with different marking probabilities p.

will also need a longer ECN-bit shift-register length to achieve higher bandwidth-utilization efficiency.

6.6 Simulation of the Proposed Scheme

To validate the analytic results and observations, we have also conducted extensive simulations. Specifically, using NetSim [31], we simulated the network performance under the proposed optimization-based multicast rate control and optimal fusion mechanism for multiple concurrent multicast connections with multiple bottlenecks. By removing the assumptions (such as the independence between different link markings) made for the modeling analysis, the simulation accurately captures the dynamics of real networks, such as the noise effect due to the randomness of network environments, and processing and queueing delays, instantaneous variations of bottleneck bandwidths, which are very difficult to handle analytically.

The simulated network is shown in Figure 6.7, and consists of 3 multicast connections

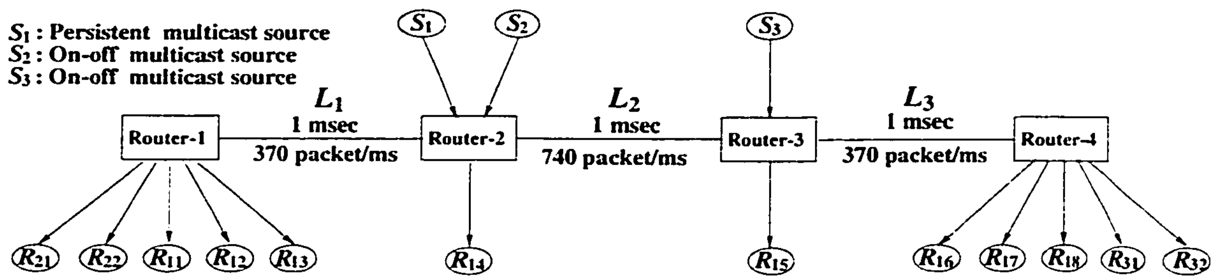


Figure 6.7: Simulation model for multiple multicast connections under the virtual M-ary feedback optimization flow control using random binary feedback.

running through 4 routers Router-1, Router-2, Router-3, Router-4, connected by 3 links L_1, L_2, L_3. s_i is the source of the i-th multicast connection, $i = 1, 2, 3$, and R_{ij} is r_i's j-th receiver. So, connections 2 and 3 share L_1 and L_3, respectively, with connection 1. r_1 is a persistent multicast source which generates the main data traffic flow. r_2 and r_3 are two periodic on-off multicast sources with on-period = 360 ms and off-period = 1011 ms, respectively, which mimic cross-traffic noises, causing the bandwidth to vary dynamically at the bottlenecks. We set L_i's bandwidth capacity μ_i as (1) $\mu_1 = \mu_3 = 370$ packets/ms; (2) $\mu_2 = 740$ packets/ms, forcing the potential bottlenecks L_1 and L_3 to emerge. Letting all links' delays be 1 ms, r_1's RTTs via R_{16}, R_{17}, R_{18} equal 4 ms which is 2 times of r_1's RTTs via R_{11}, R_{12}, R_{13}. We let r_1 start at $t = 0$, r_2 at $t = 140$ ms, and r_3 at $t = 440$ ms such that r_2 and r_3 generate the cross-traffic noises against the main data traffic flow at the potential bottlenecks L_1 and L_3 with the respective on-periods that occur alternately without any overlap in time.

We implemented the simulation model by using the NetSim event-driven simulator [31]. The flow-control parameters used in the simulation remain the same as those used in the simulation solutions shown in [2] for comparison between binary and M-ary feedback based multicast flow control. There are several types of utility functions [71, 72], but the utility function used in this paper is given by

$$U_s(r_s) \triangleq w_s \log r_s \tag{6.25}$$

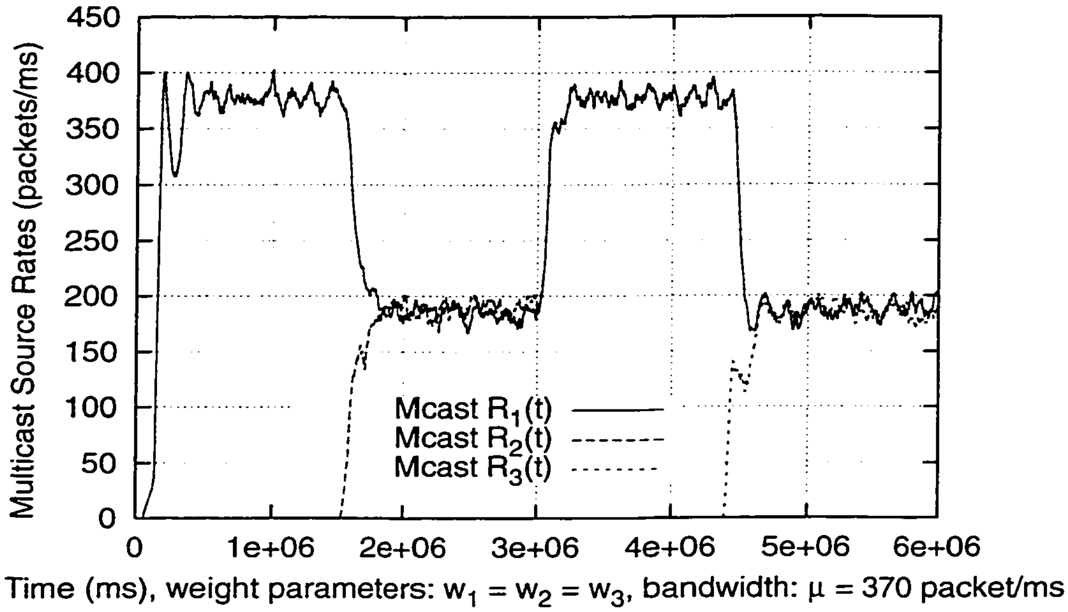


Figure 6.8: Simulated multicast source rates $R_1(t)$, $R_2(t)$, and $R_3(t)$ with same weights to receive same bandwidth share: $w_1 = w_2 = w_3$.

which leads to the maximizer of Eq. (6.9) determined by

$$\arg \max_{r_s \in I_s, s \in \mathcal{S}} \{U_s(r_s) - r_s \lambda^s_{k^*}\} = \frac{w_s}{\lambda^s_{k^*}}, \qquad \forall s \in \mathcal{S} \tag{6.26}$$

where w_s is the weight, determining the bandwidth share that source $s \in \mathcal{S}$ will receive at the bottleneck. We simulated the following two cases:

Case 1. Setting $w_1 = w_2 = w_3$, Figure 6.8 plots the rate evolution functions, $R_1(t)$, $R_2(t)$, and $R(t)$ for all multicast sources. During time $[0, 140ms]$, $R_1(t)$ converges to the target bandwidth, $\mu_2 = 370$ packets/ms at the bottleneck link L_1, which is the optimal rate the bottleneck bandwidth can support. At time $t = 140$ ms, the source s_2 enters on-state and starts data transmission, which shares the same bottleneck link L_1 with source s_1. In Figure 6.8 $R_1(t)$ and $R_2(t)$ very quickly converge to an equal share of bottle bandwidth. At time $t = 300$ ms, s_2 enters off-state and stops sending any data. Then, s_1 takes over all available bandwidth at link L_1 again. However, at time $t = 440$ ms, s_3 starts competing for bandwidth with s_1 at link L_3, i.e., the new bottleneck occurs at link L_3 which is different from the first bottleneck at link L_1. Again, Figure 6.8 shows that $R_1(t)$ and $R_3(t)$ quickly converge to the equal-share of the bandwidth of link L_3.

Figure 6.9 plots the simulated link marking probability $p_1(t)$ at the bottleneck link L_1. It shows that the bottle-link marking probability varies with bottleneck congestion level because during $[140, 300ms]$, the link marking probability is higher than the other periods. Figure 6.10 plots the multicast-tree marking probability function $p_*^{s_1}(t)$ source s_1. Comparing Figure 6.10 with Figure 6.9, we observe that during $[0, 450ms]$, $p_*^{s_1}(t)$ basically follows the pattern of $p_1(t)$. This is expected because during this period, L_1 is the only bottleneck, and thus path from s_1 to R_{11}, R_{12}, etc., is the most congested path. However, after $t=$ 300ms, $p_*^{s_1}(t)$ does not follow the pattern of $p_1(t)$ any more because the path going through L_1 is not the most congested path of $MT(s_1)$ any more. Thus, we can make the following observations:

1. The proposed virtual M-ary feedback conveys the congestion degree information by only using binary feedback.
2. The optimal feedback fusion mechanism can identify the most congested path in any given multicast tree.
3. The proposed scheme can guarantee fairness among competing multicast connections.

Case 2. For this case, we set $w_1 = 2w_2$ and only use two multicast connections $MT(s_1)$ and $MT(s_2)$ in the simulated network in Figure 6.7. The bottleneck is now located at link L_1 because s_3 does not transmit data. The simulated source rate evolution functions for both s_1 and s_2 are plotted in Figure 6.11, from which we obverse:

1. The bottleneck bandwidth is still fully-used, verifying that the proposed flow control can realize best-effort service for elastic multicast traffic sources.

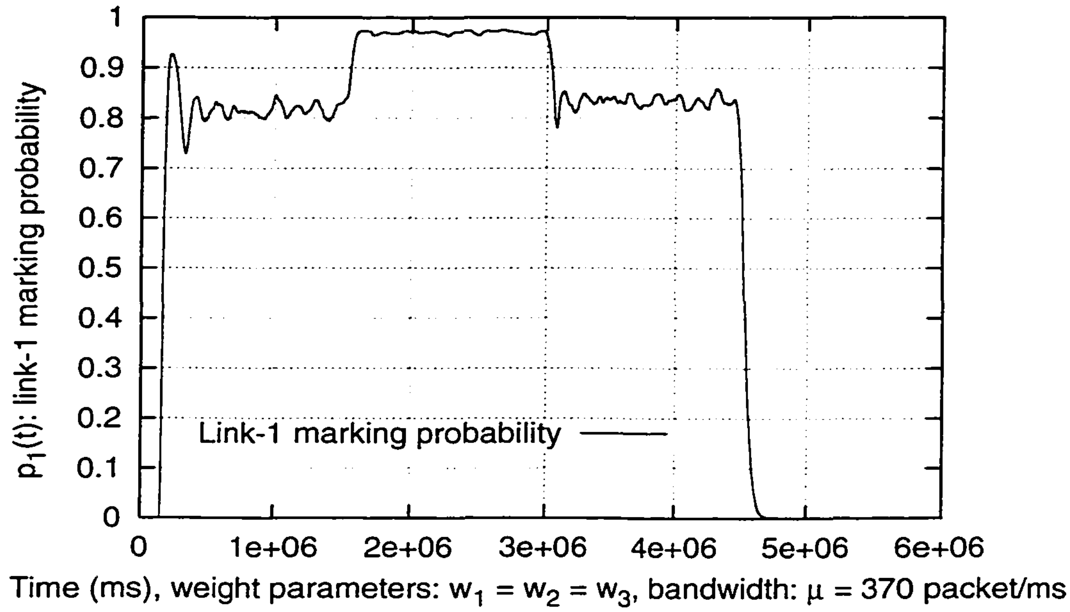


Figure 6.9: Simulated multicast link L_1 marking probability with: $w_1 = w_2 = w_3$.

2. Sources s_1 and s_2 do not share the bottleneck bandwidth evenly, but s_1 receives about 2 times as much bandwidth as s_2 because $w_1 = 2w_2$. In fact, we can arbitrarily control the bandwidth share of all multicast connections, with the same bottleneck link by properly selecting the weight factors w_s, $\forall s \in \mathcal{S}$. This means that our proposed scheme can also provide differential services to different multicast traffic connections.

6.7 Conclusion

In this chapter, we proposed and analyzed an optimization-based multicast flow control with virtual M-ary feedback for wide-area high-speed networks. Taking advantage of a history of binary feedback congestion information, the proposed scheme realized the *explicit rate control* for multicast data transmissions only by using binary random marking/congestion feedback. The proposed scheme consists of two fundamental parts: (1) the random-marking-based rate controller at the multicast source and (2) feedback fusion rules

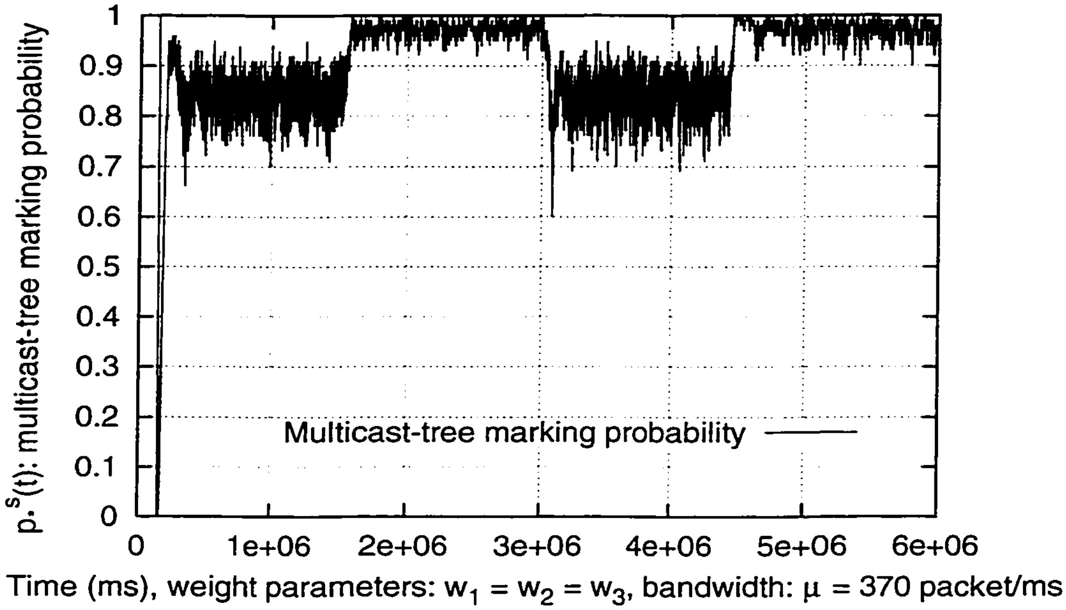


Figure 6.10: The most congested path marking probability for $MT(s_1)$ with $w_1 = w_2 = w_3$.

at all multicast routers. We used the non-linear dynamic programming to implement the multicast rate control algorithm and designed an optimal feedback fusion mechanism for consolidating the feedback signals at branch points. We developed the metrics and evaluation criteria for the design of optimal feedback ECN register size. We evaluated the proposed rate-control scheme and feedback fusion rule using both analysis and simulations.

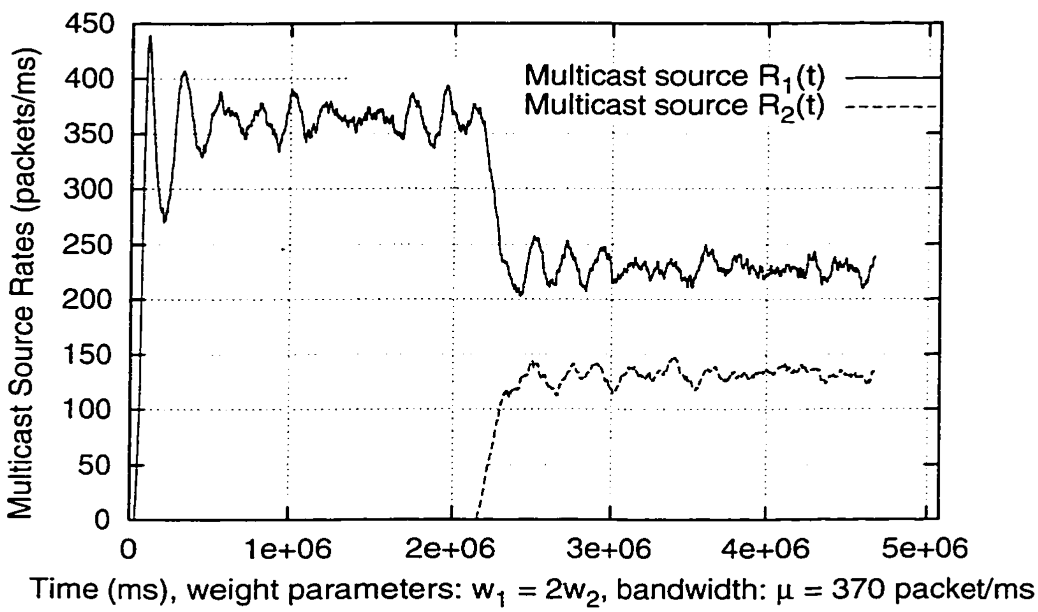


Figure 6.11: Simulated multicast source rates $R_1(t)$ and $R_2(t)$, which are given different weights to receive different bandwidth share: $w_1 = 2w_2$.

CHAPTER 7

CONCLUSION AND FUTURE WORK

We recapitulate the contributions of this dissertation, and discuss possible extensions and future directions.

7.1 Research Contributions

In this dissertation we developed flow-control protocols and modeling techniques to solve the new problems associated with multicast rate control and feedback signaling. In particular, the main contributions in this dissertation are summarized as follows.

- In Chapter 2, we designed a flow-control scheme for multicast ATM ABR services. The key of the proposed scheme is the optimal second-order rate control algorithm, called the *α-control*, developed to deal with the RTT variation resulting from dynamic "drift" of the bottleneck in a multicast tree. Using the fluid analysis, we model the proposed scheme and analyze the system dynamics for multicast ABR traffic. The analytical results and simulations show that the scheme is stable and efficient in the sense that both the source rate and bottleneck queue length rapidly converge to a small neighborhood of the designated operating point.

- To solve the feedback implosion and feedback synchronization problems imposed by multicast, in Chapter 3 we developed the *Soft Synchronization Protocol* (SSP) which consolidates the feedback RM cells at each branch that are not necessarily responses to the same forward RM cell. To evaluate the delay performance of multicast signaling protocols, we develop a balanced and unbalanced binary-tree model, by which we analyzed the feedback-delay scalability of SSP and HBH schemes.

- To evaluate the dynamic delay performance of multicast signaling protocols for the multicast flow control schemes built based on REM or RED flow control scheme, in Chapter 4 we develop a *statistical* binary-tree model to study the delay performance of a class of feedback-synchronization signaling algorithms. Applying the proposed statistical model, we derive the probability distributions for the signaling delay across the entire multicast tree and their first and second moments.

- In Chapter 5, we generalized the multicast signaling delay analysis to the cases where the congestion markings at different links are *dependent*. Specifically, we developed a Markov-chain model over link-marking states on each path in a multicast tree. We also develop a Markov-chain dependency-degree model to quantify/evaluate all possible Markov-chain dependency degrees without knowing the actual dependency degree *a priori*. Using the two models, we derived the probability distributions for the multicast bottleneck and the first and second moments of multicast signaling delay. The modeling accuracy and analytical findings have been confirmed by simulations.

- In Chapter 6, we develop a *virtual M-ary-feedback* based optimization multicast flow-control scheme, which can achieve a *fine-grained* rate control while keeping the feedback signaling traffic as low as the binary feedback. Using the *duality theory*, we first model the multicast rate control as a distributed optimization problem which decomposes the primal optimization in both aggregate utilities and constraints. We then implement the optimization by developing a distributed gradient projection algorithm and an optimal feedback ECN-bit sequence fusion mechanism.

7.2 Future Research Directions

The work performed in this dissertation has revealed several promising research issues that are worth further investigation:

Integrated Error Control with Multicast Flow Control. As in unicast, error control is also important for reliable multicast data transmissions over lossy networks. While error control in principle can be treated separately from flow control [8], they are closely related to each other and often blended together in implementations, such as in TCP [1]. There are two major new challenges associated with multicast error control. The first problem is that packet retransmissions from the sender are delivered to the entire multicast group, not only to receivers which did not receive the packets. This causes unwanted packet processing at receivers which have already received the packets, and wastes network bandwidth on the links on paths to these receivers. Secondly, when the number of receivers is large and the lossy probability is high, the probability of at least one receiver losing the packet is very high. This means that almost every packet is likely to be retransmitted, perhaps several times. Subsequently, the multicast sender may have to conduct a large number of retransmissions to ensure reliable delivery to potentially thousands of receivers. Thus, the sender and the links in its proximity are likely to become bottlenecks, resulting in a significant degradation of the overall throughput.

The first problem can be overcome by the *retransmission scoping* [83] technique which retransmits the packets only to the sub-multicast tree that did not see the packet, while the second problem can be solved by the *distributed loss recovery* [82,84,85] which allows efficient loss recovery from the point of loss using a nearby repair agent (a designated server at router or receiver) for local multicast of retransmissions. The multicast flow control scheme proposed in Chapter 2, together with the SSP developed in Chapter 3, can be enhanced for error control by including the retransmission scoping capability to implement the scalable reliable multicast flow control. This is because the proposed flow-control framework and the SSP make it very easy to implement the selective multicast of retransmissions for lost packets. On the other hand, the multicast flow-control scheme developed in Chapter 6 can be extended to implement the distributed loss recovery mechanism since the optimal feedback fusion mechanism installed at each multicast router can easily determine the most congested sub-multicast tree and thus cache packets in advance to subcast (multicast on the subtree below the router connected to the repair server) the retransmissions for lost packets if needed.

Multiple-Rate Multicast Flow Control. In this dissertation, we have mainly focused on the single rate multicast flow control, where all receivers in the multicast tree receive data or information at a single identical rate. The single-rate based scheme is cost-effective in implementations and resource management, and suitable for sender-oriented multicast flow control. There is another category of multicast flow control where different receivers within a multicast session can receive the data packet at different rates [73, 74, 86–89]. This can be accomplished by layering data among several multicast groups and allowing each receiver to independently determine the subset of layers (i.e., multicast groups) it joins or a single data layering with router-filtering with different branching rates at multicast routers. Protocols that use a layered approach to support multicast applications range from live multimedia streams to reliable data transfer. The common property of these protocols is that the transmission rate to each receiver is constrained only by the bandwidth availability on the receiver's own data-path from the data source, and is not limited by other receivers' rate limitations in the same multicast session. This ensures the *intra-session* fairness such that each receiver can receive service at a rate commensurate with its capacity and the capacity of the path leading to it from the source, independently of the capacities of the other receivers of the same multicast session. The virtual M-ary feedback optimization multicast flow-control scheme proposed in Chapter 6 provides the basic facilities and mechanisms to support multiple-rate multicast flow control through either data layering or router filtering techniques. However, how to implement the multiple-rate multicast flow control by extending the proposed scheme to support both live multimedia streams and reliable data transfer remains an interesting and challenging research topic.

Multicast Flow Control over Wireless Networks. The anticipated multicast streaming and data applications in mobile computing environments impose the stergent requirement on reliability and traffic management strategies [90–104]. Error and congestion control for continuous media [105] (e.g., audio and video) multicast over wireless networks remains a challenging problemc̄iteTowsley:85. Most conventional error-control techniques based on ARQ (Automatic Repeat reQuest) are not applicable for multicast over the wireless/mobile environment because: (1) retransmission-based ARQ does not scale well to multicast group with a large number of receivers; (2) communication over wireless links is highly asymmetric and receivers can move frequently; (3) the processing capacities and loss rates are highly heterogeneous among the multicast receivers over the different wireless links; (4) in wireless networks, uplink communication is very expensive due to power consumption/limitation, and thus is often constrained.

A different approach often used to improve reliability is FEC (Forward Error Correction) which adds the controlled redundancy to the data stream such that a receiver to recover from packet losses without asking the source for retransmissions. FEC is not only attractive for error control in multicast over wireless networks since it does not impose the above-mentioned problems, but also provides an efficient congestion-control approach for multicast as recently proposed in [76]. The implementation of the FEC-based multicast flow control needs a router-assisted flow-control scheme, which can be well supported by the virtual M-ary feedback optimization multicast flow framework proposed in Chapter 6. Thus, how to properly integrate the flow control with error control in a unified FEC-based framework setup using the proposed virtual M-ary feedback optimization multicast flow control over wireless networks is an another interesting research direction to take.

APPENDICES

APPENDIX A

PROOF OF THEOREM 2.4.1

Proof. Using the fluid-modeling results on the multicast-tree bottleneck described in Section 2.5, for $(\alpha, \tau) \in \Omega$ we have (see the derivations of Eqs. (2.15) and (2.17))

$$Q_{max}(\alpha, \tau) = \int_0^{T_{max}} \alpha t \; dt + \int_0^{T_d} \left(R_{max} e^{-(1-\beta)\frac{t}{\Delta}} - \mu \right) dt \tag{A.1}$$

$$= \frac{\alpha}{2} T_{max}^2 + \frac{\Delta}{1-\beta} \left(\alpha \, T_{max} + \mu \log \frac{\mu}{R_{max}} \right) \tag{A.2}$$

where μ is the multicast-tree bottleneck target bandwidth, and

$$T_{max} = \tau + \sqrt{\frac{2Q_h}{\alpha}}, \tag{A.3}$$

$$R_{max} = \mu + \alpha \left(\tau + \sqrt{\frac{2Q_h}{\alpha}} \right), \tag{A.4}$$

$$T_d = \frac{\Delta}{(1-\beta)} \log \left[1 + \frac{\alpha}{\mu} \left(\tau + \sqrt{\frac{2Q_h}{\alpha}} \right) \right]. \tag{A.5}$$

On the other hand, Q_{max} is also equal to the area between $R(t)$ and μ over the time interval of $T_{max} + T_d$, and is upper-bounded by the area of its circumscribed triangle $\triangle ABC$ as shown in Figure A.1. Thus, we have

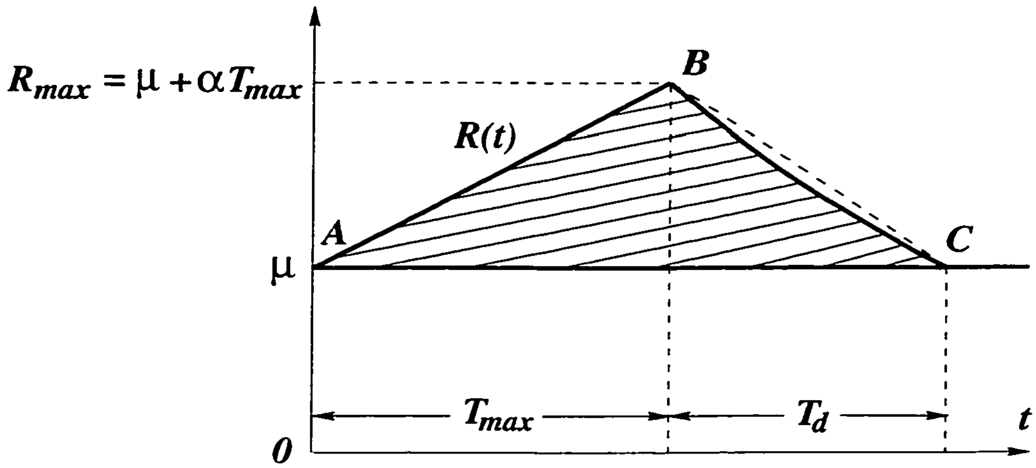


Figure A.1: Q_{max} (shaded area) is upper-bounded by the area of $\triangle ABC$.

$$
\begin{aligned}
&Q_{max}(\alpha,\tau)\\
&\le \frac{1}{2}(\alpha T_{max})(T_{max}+T_d)\\
&= \frac{1}{2}\left[\alpha\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)^2+\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\left(\frac{\alpha\Delta}{1-\beta}\log\left[1+\frac{\alpha}{\mu}\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\right]\right)\right]\\
&= \frac{1}{2}\left[\alpha\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)^2+\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\left(\mu\log\left[1+\frac{\alpha}{\mu}\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\right]\right)\right] && \text{(A.6)}\\
&\le \frac{1}{2}\left[\alpha\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)^2+\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\left(\mu\left[\frac{\alpha}{\mu}\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\right]\right)\right] && \text{(A.7)}\\
&= (\tau\sqrt{\alpha}+\sqrt{2Q_h})^2 && \text{(A.8)}
\end{aligned}
$$

Since $\alpha > 0$ due to $(\alpha,\tau) \in \Omega$, in Eq. (A.6) we can apply the given condition equality (constraint) of $\alpha\left(\frac{\Delta}{1-\beta}\right) = \mu$ which is set to balance the increasing and decreasing speeds of $R(t)$ (for the details, please see [28]). Eq. (A.7) is due to the fact that $\log x \le x - 1$ (Note: $\log x \approx x - 1$ for x close to 1. So, the derived upper bound becomes tighter if $\left[1+\frac{\alpha}{\mu}\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\right] = \frac{1}{\mu}R_{max}$ is close to 1, i.e., $\mu < R_{max} = \mu + \alpha\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right) \ll 2\mu$, or equivalently $1 < \frac{1}{\mu}R_{max} \ll 2$, which is the typical operating regime for the proposed α-control based flow control scheme since α is small under the α-control for the given finite buffer capacity C_{max}). Equation (A.8) yields the upper bound derived by Eq. (2.4), completing the proof. ■

APPENDIX B

PROOF OF THEOREM 2.4.2

Proof. <u>Claim 1:</u> Let $K \triangleq \tau\sqrt{\alpha}$ which is a positive real number for $(\alpha, \tau) \in \Omega$. Define a real-valued function $\zeta(K) = \zeta(\tau\sqrt{\alpha}) \triangleq (K + \sqrt{2Q_h})^2$, which is the upper-bound function of $Q_{max}(\alpha, \tau)$ obtained from Eq. (A.8). Thus, by Theorem 2.4.1 we have $\zeta(K) \geq Q_{max}(\alpha, \tau)$ for $(\alpha, \tau) \in \Omega$ and further

$$Q_{max}(\alpha, \tau) \leq \zeta(K) = [K^2 + 2\sqrt{2Q_h}K + (2Q_h - C_{max})] + C_{max}. \tag{B.1}$$

Since $C_{max} > 2Q_h$ and $\zeta(K)$ is a continuous and monotonically-increasing function of K, $\exists\ K > 0$ such that

$$K^2 + 2\sqrt{2Q_h}K < (C_{max} - 2Q_h), \qquad \text{i.e.,} \qquad [K^2 + 2\sqrt{2Q_h}K + (2Q_h - C_{max})] < 0. \tag{B.2}$$

and $\{(\alpha, \tau) \mid \tau\sqrt{\alpha} \leq K,\ (\alpha, \tau) \in \Omega\} \neq \emptyset$. Thus, $\forall\ (\alpha, \tau) \in \{(\alpha, \tau) \mid \tau\sqrt{\alpha} \leq K,\ (\alpha, \tau) \in \Omega\}$ where K is specified by Eq. (B.2), by Eqs. (B.1) and (B.2), we get

$$Q_{max}(\alpha, \tau) \leq [K^2 + 2\sqrt{2Q_h}K + (2Q_h - C_{max})] + C_{max} < C_{max}, \tag{B.3}$$

which implies $(\alpha, \tau) \in \mathcal{F}$, thus $\mathcal{F} \neq \emptyset$.

<u>Claim 2:</u> To obtain a tight lower bound for $\mathcal{L}$, we set $Q_{max}(\alpha, \tau)$'s upper-bound function $\zeta(K)$ equal to C_{max}, i.e.,

$$Q_{max}(\alpha, \tau) \leq \zeta(K) = K^2 + 2\sqrt{2Q_h}K + 2Q_h = C_{max}, \tag{B.4}$$

which reduces to a quadratic equation: $K^2 + 2\sqrt{2Q_h}K + (2Q_h - C_{max}) = 0$. Solving this for K and taking the positive root, $K_\ell = \sqrt{C_{max}} - \sqrt{2Q_h} > 0$ since $C_{max} > 2Q_h$. By Eq. (B.4), $(\alpha, \tau) \in \mathcal{F}$, $\forall\ (\alpha, \tau) \in \{(\alpha, \tau) \mid \tau\sqrt{\alpha} \leq K_\ell,\ (\alpha, \tau) \in \Omega\}$, implying that all points located below or on the curve of function $K_\ell = \tau\sqrt{\alpha} \notin \mathcal{L}$. Thus, $\mathcal{L}$ is lower bounded by the function of $\zeta(K) = C_{max}$ or $K_\ell = \tau\sqrt{\alpha} = \sqrt{C_{max}} - \sqrt{2Q_h}$, completing the proof. ■

APPENDIX C

PROOF OF THEOREM 2.4.3

Proof. Claim 1: We prove this claim by considering the following two cases depending upon the range of the initial value of rate-gain parameter α_0.

Case 1. $\alpha_0 \leq \alpha_{goal}$: $Q_{max}(\alpha)$ is a monotonically-increasing function of α, $\alpha_0 \leq \alpha_{goal} \Rightarrow Q_{max}^{(0)} = Q_{max}(\alpha_0) \leq Q_{goal} = Q_{max}(\alpha_{goal})$. Applying the α-control law, $Q_{max}^{(n)}$ monotonically increases from $Q_{max}^{(0)}$ towards Q_{goal} with an increase-step size p. When $Q_{max}^{(n)}$ the first time becomes larger than Q_{goal} at $n = n^*$, i.e., $\alpha_0 + n^*p = \alpha_{n^*} > \alpha_{goal}$, the source detects $BCI(n^* - 1, n^*) = (0, 1)$, and then reduces α_n exponentially by setting $\alpha_{n^*+1} = q\alpha_{n^*}$ $(0 < q < 1)$. We want to prove the following fact:

$$Q_{max}(\alpha_{n^*+1}) = Q_{max}(q\alpha_{n^*}) \leq Q_{goal} \tag{C.1}$$

Since $(\tau\sqrt{\alpha_{goal}} + \sqrt{2Q_h})^2 \geq Q_{max}(\alpha_{goal}) = Q_{goal}$ by Theorem 2.4.1, we have

$$\left(\frac{\sqrt{Q_{goal}} - \sqrt{2Q_h}}{\tau}\right)^2 \leq \alpha_{goal}. \tag{C.2}$$

But, since

$$p \leq \left(\frac{1-q}{q}\right)\left(\frac{\sqrt{Q_{goal}} - \sqrt{2Q_h}}{\tau}\right)^2, \tag{C.3}$$

we obtain

$$p \leq \left(\frac{1-q}{q}\right)\alpha_{goal} \tag{C.4}$$

which reduces to

$$q(\alpha_{goal} + p) \leq \alpha_{goal}. \tag{C.5}$$

On the other hand, due to $\alpha_{n^*-1} \leq \alpha_{goal}$, we have $q(\alpha_{n^*-1} + p) \leq q(\alpha_{goal} + p)$. Because $\alpha_{n^*} = \alpha_{n^*-1} + p$, $q(\alpha_{n^*-1} + p) \leq q(\alpha_{goal} + p)$, and $q(\alpha_{goal} + p) \leq \alpha_{goal}$, we get

$$\alpha_{n^*+1} = q\alpha_{n^*} = q(\alpha_{n^*-1} + p) \leq q(\alpha_{goal} + p) \leq \alpha_{goal} \tag{C.6}$$

Thus, $Q_{max}(\alpha_{n^*+1}) = Q_{max}(q\alpha_{n^*}) \leq Q_{max}(\alpha_{goal}) = Q_{goal}$, which is Eq. (C.1). Due to Eq. (C.1), $BCI(n^*, n^* + 1) = (1, 0)$. Applying the α-control law, we get $\alpha_{n^*+2} = \alpha_{n^*+1}/q$. But $\alpha_{n^*+1} = q\alpha_{n^*}$, giving $\alpha_{n^*+2} = q\alpha_{n^*}/q = \alpha_{n^*} > \alpha_{goal}$; thus, $BCI(n^* + 1, n^* + 2) = BCI(0, 1)$. Applying the α-control law again, $\alpha_{n^*+3} = q\alpha_{n^*+2} = q\alpha_{n^*} = \alpha_{n^*+1}$. But by Eq. (C.6), $\alpha_{n^*+3} = q\alpha_{n^*} \leq \alpha_{goal}$, and thus $BCI(n^* + 2, n^* + 3) = (1, 0)$. Repeating the above procedure, we get $\forall\ k \in \{0, 1, 2, \cdots, \}$

$$\begin{cases} \alpha_{n^*+(2k+1)} = \alpha_{n^*+1} = q\alpha_{n^*} \leq \alpha_{goal} \\ \alpha_{n^*+2k} = \alpha_{n^*} > \alpha_{goal}; \end{cases} \tag{C.7}$$

implying that

$$\begin{aligned} &BCI(0, 1, 2, 3, \cdots, n^* - 1, n^*, n^* + 1, n^* + 2, n^* + 3, \cdots) \\ &= (0, 0, 0, 0, \cdots, 0, 1, 0, 1, 0, \cdots). \end{aligned} \tag{C.8}$$

By Definition 2.4.3, $Q_{max}^{(n)}$ monotonically converges to Q_{goal}'s neighborhood $\{Q_{goal}^{l}, Q_{goal}^{h}\}$. In addition, in the equilibrium state,

$$\begin{cases} Q_{max}(q\alpha_{n^*}) = Q_{max}(q(n^*p + \alpha_0)) = \max_{n\in\{0,1,2,\cdots\}} \{Q_{max}^{(n)} \mid Q_{max}^{(n)} \leq Q_{goal}\}; \\ Q_{max}(\alpha_{n^*}) = Q_{max}(n^*p + \alpha_0) = \min_{n\in\{0,1,2,\cdots\}} \{Q_{max}^{(n)} \mid Q_{max}^{(n)} > Q_{goal}\}. \end{cases} \tag{C.9}$$

Thus, by Definition 2.4.2, $Q_{goal}^{l} = Q_{max}(q(n^*p + \alpha_0))$ and $Q_{goal}^{h} = Q_{max}(n^*p + \alpha_0)$.

<u>**Case 2.** $\alpha_0 > \alpha_{goal}$:</u> Since $Q_{max}^{(0)} = Q_{max}(\alpha_0) > Q_{goal} = Q_{max}(\alpha_{goal})$, applying the α-control law, $Q_{max}^{(n)}$ monotonically decreases from $Q_{max}^{(0)}$ towards Q_{goal} with a factor q $(0 <$

$q < 1$). When $Q_{max}^{(n)} \le Q_{goal}$ for the first time at $n = n^*$, i.e., $q^{n^*}\alpha_0 = \alpha_{n^*} \le \alpha_{goal}$, the source detects $BCI(n^* - 1, n^*) = (1, 0)$. Applying the α-control law, we get $\alpha_{n^*+1} = \alpha_{n^*}/q = \alpha_{n^*-1} > \alpha_{goal}$, and thus $BCI(n^*, n^*+1) = (0, 1)$. By α-control, $\alpha_{n^*+2} = q\alpha_{n^*+1} = q(\alpha_{n^*}/q) = \alpha_{n^*} \le \alpha_{goal}$, and thus $BCI(n^* + 1, n^* + 2) = (1, 0)$. Applying the α-control law again, we get $\alpha_{n^*+3} = \alpha_{n^*+2}/q = \alpha_{n^*+1} > \alpha_{goal}$, and thus $BCI(n^* + 2, n^* + 3) = (0, 1)$. Repeat the above deducing procedure, we have $\forall\ k \in \{0, 1, 2, \cdots, \}$

$$\begin{cases} \alpha_{n^*+(2k+1)} = \alpha_{n^*+1} = \alpha_{n^*}/q > \alpha_{goal} \\ \alpha_{n^*+2k} = \alpha_{n^*} \le \alpha_{goal}; \end{cases} \tag{C.10}$$

implying that

$$\begin{aligned} &BCI(0, 1, 2, 3, \cdots, n^* - 1, n^*, n^* + 1, n^* + 2, n^* + 3, \cdots) \\ &\quad = (1, 1, 1, 1, \cdots, 1, 0, 1, 0, 1, \cdots). \end{aligned} \tag{C.11}$$

Therefore, by Definition 2.4.3, $Q_{max}^{(n)}$ monotonically converges to Q_{goal}'s neighborhood $\{Q_{goal}^l, Q_{goal}^h\}$. In addition, in the equilibrium state,

$$\begin{cases} Q_{max}(\alpha_{n^*}) = Q_{max}(q^{n^*}\alpha_0) = \max_{n\in\{0,1,2,\cdots\}}\ \{Q_{max}^{(n)} \mid Q_{max}^{(n)} \le Q_{goal}\}; \\ Q_{max}(\alpha_{n^*}/q) = Q_{max}(q^{(n^*-1)}\alpha_0) = \min_{n\in\{0,1,2,\cdots\}}\ \{Q_{max}^{(n)} \mid Q_{max}^{(n)} > Q_{goal}\}. \end{cases} \tag{C.12}$$

Thus, by Definition 2.4.2, $Q_{goal}^l = Q_{max}(q^{n^*}\alpha_0)$ and $Q_{goal}^h = Q_{max}(q^{(n^*-1)}\alpha_0)$.

<u>**Claim 2:**</u> Since $0 < q < 1$ and $p \le \left(\frac{1-q}{q}\right)\left(\frac{\sqrt{Q_{goal}} - \sqrt{2Q_h}}{\tau}\right)^2$, by Claim 1 of Theorem 2.4.3, $Q_{max}^{(n)}$ is guaranteed to converge to Q_{goal}'s neighborhood in the equilibrium state. Define maximum-queue-length upper-bound error function for $(\alpha, \tau) \in \mathcal{F}$ by

$$\gamma(\alpha, \tau) \triangleq \zeta(\tau\sqrt{\alpha}) - Q_{max}(\alpha, \tau) = (\tau\sqrt{\alpha} + \sqrt{2Q_h})^2 - Q_{max}(\alpha, \tau), \qquad (\alpha, \tau) \in \mathcal{F} \tag{C.13}$$

which is a non-negative real-valued function since $Q_{max}(\alpha, \tau) \le \zeta(\tau\sqrt{\alpha})$. According to Lemma D.1.1 given in Appendix D, which is also verified in Figure 2.4, and because $\alpha_{goal}^h \ge$

α_{goal}, we have $\gamma(\alpha^h_{goal}, \tau) - \gamma(\alpha_{goal}, \tau) \geq 0$, which leads to:

$$
\begin{aligned}
& Q^h_{goal} - Q_{goal} \\
& \quad \leq \left[Q^h_{goal} - Q_{goal}\right] + \left[\gamma(\alpha^h_{goal}, \tau) - \gamma(\alpha_{goal}, \tau)\right] \\
& \quad = Q^h_{goal} - Q_{goal} + \left[\left(\tau\sqrt{\alpha^h_{goal}} + \sqrt{2Q_h}\right)^2 - Q^h_{goal}\right] - \left[\left(\tau\sqrt{\alpha_{goal}} + \sqrt{2Q_h}\right)^2 - Q_{goal}\right] \\
& \quad = \tau^2\left(\alpha^h_{goal} - \alpha_{goal}\right) + \tau\sqrt{8Q_h}\left(\sqrt{\alpha^h_{goal}} - \sqrt{\alpha_{goal}}\right) \\
& \quad \leq \tau^2\left(\frac{1}{q}\alpha_{goal} - \alpha_{goal}\right) + \tau\sqrt{8Q_h}\left(\sqrt{\frac{1}{q}\alpha_{goal}} - \sqrt{\alpha_{goal}}\right) & \text{(C.14)} \\
& \quad = \tau^2\alpha_{goal}\left(\frac{1}{q} - 1\right) + \tau\sqrt{8\alpha_{goal}Q_h}\left(\frac{1}{\sqrt{q}} - 1\right) & \text{(C.15)}
\end{aligned}
$$

where Eq. (C.14) is due to the inequality $\alpha^h_{goal} \leq \frac{1}{q}\alpha_{goal}$ that resulted from the α-control law. This proves Eq. (2.10). Likewise, because $\alpha_{goal} \geq \alpha^l_{goal}$, which results in $\gamma(\alpha_{goal}, \tau) - \gamma(\alpha^l_{goal}, \tau) \geq 0$ due to Lemma D.1.1 given in Appendix D ensuring $\gamma(\alpha, \tau)$ to be a monotonically increasing function of α, we obtain

$$
\begin{aligned}
& Q_{goal} - Q^l_{goal} \\
& \quad \leq \left[Q_{goal} - Q^l_{goal}\right] + \left[\gamma(\alpha_{goal}, \tau) - \gamma(\alpha^l_{goal}, \tau)\right] \\
& \quad = Q_{goal} - Q^l_{goal} + \left[\left(\tau\sqrt{\alpha_{goal}} + \sqrt{2Q_h}\right)^2 - Q_{goal}\right] - \left[\left(\tau\sqrt{\alpha^l_{goal}} + \sqrt{2Q_h}\right)^2 - Q^l_{goal}\right] \\
& \quad = \tau^2\left(\alpha_{goal} - \alpha^l_{goal}\right) + \tau\sqrt{8Q_h}\left(\sqrt{\alpha_{goal}} - \sqrt{\alpha^l_{goal}}\right) \\
& \quad \leq \tau^2\left(\alpha_{goal} - q\alpha_{goal}\right) + \tau\sqrt{8Q_h}\left(\sqrt{\alpha_{goal}} - \sqrt{q\alpha_{goal}}\right) & \text{(C.16)} \\
& \quad = \tau^2\alpha_{goal}(1 - q) + \tau\sqrt{8\alpha_{goal}Q_h}(1 - \sqrt{q}) & \text{(C.17)}
\end{aligned}
$$

where Eq. (C.16) is due to the fact that $\alpha^l_{goal} \geq q\alpha_{goal}$ resulting from the α-control law. This proves Eq. (2.11). Adding both sides of Eq. (C.15) and those of Eq. (C.17), Eq. (2.12) follows, which completes the proof. ■

APPENDIX D

PROOF OF LEMMA D.1.1

D.1 Maximum Queue-Length Upper-Bound Error Function Monotonicity Lemma

Lemma D.1.1 *The maximum-queue-length upper-bound error function* $\gamma(\alpha,\tau) = \zeta(\tau\sqrt{\alpha}) - Q_{max}(\alpha,\tau) = (\tau\sqrt{\alpha}+\sqrt{2Q_h})^2 - Q_{max}(\alpha,\tau)$ *defined in Eq. (C.13), is a strictly monotonic-increasing function with respect to* α, $\forall\alpha > 0$ *and* $(\alpha,\tau) \in \mathcal{F}$. ■

D.2 Proof of the Maximum Queue-Length Upper-Bound Error Function Monotonicity Lemma (Lemma D.1.1)

Proof. Since $\gamma(\alpha,\tau)$ is defined only for $(\alpha,\tau) \in \mathcal{F}$, we only need to consider $(\alpha,\tau) \in \mathcal{F} \subset \Omega$ where $\gamma(\alpha,\tau)$ is continuous and differentiable, and thus we can take a partial derivative over $\gamma(\alpha,\tau)$ with respect to α as follows:

$$\frac{\partial\gamma(\alpha,\tau)}{\partial\alpha} = \frac{\partial\zeta(\alpha,\tau)}{\partial\alpha} - \frac{\partial Q_{max}(\alpha,\tau)}{\partial\alpha} \tag{D.1}$$

where

$$
\begin{aligned}
\frac{\partial \zeta(\alpha,\tau)}{\partial \alpha} &= \tau^2+\tau\sqrt{\frac{2Q_h}{\alpha}}, && \text{(D.2)}\\
\frac{\partial Q_{max}(\alpha,\tau)}{\partial \alpha} &= \frac{1}{2}\tau^2+\tau\sqrt{\frac{2Q_h}{\alpha}}-\tau\sqrt{\frac{Q_h}{2\alpha}}-\mu\sqrt{\frac{Q_h}{2}}\alpha^{-\frac{3}{2}}+\frac{\mu^2}{\alpha^2}\log\left(1+\frac{\alpha}{\mu}\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\right)\\
&\quad -\frac{\mu^2\left(\tau+\sqrt{\frac{Q_h}{2\alpha}}\right)}{\mu\alpha+\alpha^2\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)}. && \text{(D.3)}
\end{aligned}
$$

Note that again, we use fact that $\mu=\frac{\Delta\alpha}{1-\beta}$ in derivations of $\frac{\partial Q_{max}(\alpha,\tau)}{\partial\alpha}$ in Eq. (D.3). Thus, we obtain

$$
\begin{aligned}
\frac{\partial \gamma(\alpha,\tau)}{\partial \alpha} &= \frac{\tau^2}{2}+\tau\sqrt{\frac{Q_h}{2\alpha}}+\mu\sqrt{\frac{Q_h}{2}}\alpha^{-\frac{3}{2}}-\frac{\mu^2}{\alpha^2}\log\left[1+\frac{\alpha}{\mu}\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\right]\\
&\quad +\frac{\mu^2\left(\tau+\sqrt{\frac{Q_h}{2\alpha}}\right)}{\mu\alpha+\alpha^2\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)}. && \text{(D.4)}
\end{aligned}
$$

Using Eq. (D.4), we define a new real-valued function $\varphi(\alpha,\tau)$ as follows:

$$
\begin{aligned}
\varphi(\alpha,\tau) &\triangleq \left(\frac{\alpha^2}{\mu^2}\right)\frac{\partial\gamma(\alpha,\tau)}{\partial\alpha}\\
&= \frac{1}{2}\left(\frac{\tau\alpha}{\mu}\right)^2+\frac{\tau}{\mu^2}\sqrt{\frac{Q_h}{2}}\alpha^{\frac{3}{2}}+\frac{1}{\mu}\sqrt{\frac{Q_h\alpha}{2}}-\log\left(\frac{\mu+\tau\alpha+\sqrt{2Q_h\alpha}}{\mu}\right)\\
&\quad +\frac{\alpha\left(\tau+\sqrt{\frac{Q_h}{2\alpha}}\right)}{\mu+\alpha\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)} && \text{(D.5)}
\end{aligned}
$$

Taking partial derivative with respect to α on both sides of Eq. (D.5), we obtain

$$
\begin{aligned}
\frac{\partial\varphi(\alpha,\tau)}{\partial\alpha} &= \frac{1}{(\mu+\sqrt{2Q_h\alpha}+\tau\alpha)^2}\left[\frac{Q_h^{\frac{3}{2}}}{\mu}\sqrt{\frac{\alpha}{2}}+\frac{4\tau Q_h}{\mu}\alpha+\alpha^2\left(\frac{2\tau^3}{\mu}+\frac{5\tau^2Q_h}{\mu^2}\right)\right.\\
&\quad +\alpha^{\frac{3}{2}}\left(\frac{2\tau^2}{\mu}\sqrt{2Q_h}+\frac{3\tau^3}{\mu}\sqrt{\frac{Q_h}{2}}+\frac{3\tau Q_h^{\frac{3}{2}}}{\sqrt{2}\mu^2}+\frac{\tau^2}{2\mu}\sqrt{\frac{Q_h}{2}}\right)\\
&\quad \left.+\alpha^{\frac{5}{2}}\left(\frac{2\tau^3}{\mu^2}\sqrt{2Q_h}+\frac{3\tau^3}{2\mu^2}\sqrt{\frac{Q_h}{2}}\right)+\frac{\tau^4}{\mu^2}\alpha^3\right]>0, && \text{(D.6)}
\end{aligned}
$$

That is, Eq. (D.6) proves:

$$
\frac{\partial\varphi(\alpha,\tau)}{\partial\alpha}>0 \quad \forall\alpha>0,\ (\alpha,\tau)\in\mathcal{F}, \qquad \text{(D.7)}
$$

implying that $\varphi(\alpha, \tau)$ is a strictly monotonic-increasing function with respect to α, $\forall\alpha > 0$ and $(\alpha, \tau) \in \mathcal{F}$. On the other hand, note

$$\varphi(\alpha, \tau) \mid_{\alpha=0} = 0. \tag{D.8}$$

Combining Eq. (D.7) and Eq. (D.8), it follows that $\varphi(\alpha, \tau) > 0$, $\forall\alpha > 0$ and $(\alpha, \tau) \in \mathcal{F}$, and that is for $\forall\alpha > 0$ and $(\alpha, \tau) \in \mathcal{F}$, we have

$$\varphi(\alpha, \tau) = \left(\frac{\alpha^2}{\mu^2}\right) \frac{\partial\gamma(\alpha, \tau)}{\partial\alpha} > 0. \tag{D.9}$$

Reducing Eq. (D.9), we obtain

$$\frac{\partial\gamma(\alpha, \tau)}{\partial\alpha} > 0, \quad \forall\alpha > 0 \text{ and } (\alpha, \tau) \in \mathcal{F}, \tag{D.10}$$

which completes the proof of Lemma D.1.1. ■

APPENDIX E

PROOF OF THEOREM 2.5.1

Proof. We also need to prove this theorem by considering the following two cases, which correspond to the first and second parts of Eq. (2.23), respectively

Case 1. $\alpha_0 > \alpha_{goal}$: Let $\widetilde{\alpha^l_{goal}}$ correspond to the new $\widetilde{Q^l_{goal}} = Q_{max}(\widetilde{\alpha^l_{goal}})$. From Eq. (2.8), we have $\widetilde{\alpha^l_{goal}} = q^{n^*}\alpha_0$, which reduces to

$$n^* = \frac{\log \frac{\alpha_0}{\widetilde{\alpha^l_{goal}}}}{\log \frac{1}{q}} \geq \frac{\log \frac{\alpha_0}{\widetilde{\alpha_{goal}}}}{\log \frac{1}{q}} = \frac{\log \frac{\widetilde{\alpha_{goal}}}{\alpha_0}}{\log q} \tag{E.1}$$

where

$$\frac{\log \frac{\alpha_0}{\widetilde{\alpha^l_{goal}}}}{\log \frac{1}{q}} \geq \frac{\log \frac{\alpha_0}{\widetilde{\alpha_{goal}}}}{\log \frac{1}{q}} \tag{E.2}$$

is due to $\widetilde{\alpha_{goal}} \geq \widetilde{\alpha^l_{goal}}$. But since $q\widetilde{\alpha_{goal}} < \widetilde{\alpha^l_{goal}}$, that is

$$\frac{\log \frac{\alpha_0}{\widetilde{\alpha^l_{goal}}}}{\log \frac{1}{q}} - \frac{\log \frac{\alpha_0}{\widetilde{\alpha_{goal}}}}{\log \frac{1}{q}} < 1, \tag{E.3}$$

we obtain

$$\frac{\log \frac{\widetilde{\alpha_{goal}}}{\alpha_0}}{\log q} \leq n^* = \frac{\log \frac{\alpha_0}{\widetilde{\alpha^l_{goal}}}}{\log \frac{1}{q}} < 1 + \frac{\log \frac{\widetilde{\alpha_{goal}}}{\alpha_0}}{\log q} \qquad \Longrightarrow \qquad n^* = \left\lceil \log\left(\frac{\widetilde{\alpha_{goal}}}{\alpha_0}\right) / \log q \right\rceil, \tag{E.4}$$

since n^* must be an integer. By Definition 2.4.3, $N = n^* - 1$ for $\alpha > \widetilde{\alpha_{goal}}$. Thus we have

$$N = n^* - 1 = \left\lfloor \log\left(\frac{\widetilde{\alpha_{goal}}}{\alpha_0}\right) / \log q \right\rfloor. \tag{E.5}$$

Case 2. $\alpha_0 \leq \alpha_{goal}$: Let $\widetilde{\alpha_{goal}^h}$ correspond to the new $\widetilde{Q_{goal}^h} = Q_{max}(\widetilde{\alpha_{goal}^h})$. From Eq. (2.9), we have $\widetilde{\alpha_{goal}^h} = n^* p + \alpha_0$, which reduces to

$$n^* = \frac{\widetilde{\alpha_{goal}^h} - \alpha_0}{p} \geq \frac{\widetilde{\alpha_{goal}} - \alpha_0}{p} \tag{E.6}$$

where the inequality in Eq. (E.6) is due to $\widetilde{\alpha_{goal}} \leq \widetilde{\alpha_{goal}^h}$. Since $\widetilde{\alpha_{goal}^h} - \widetilde{\alpha_{goal}} < p$, i.e., $\frac{\widetilde{\alpha_{goal}^h} - \alpha_0}{p} - \frac{\widetilde{\alpha_{goal}} - \alpha_0}{p} < 1$, we get

$$\frac{\widetilde{\alpha_{goal}} - \alpha_0}{p} \leq n^* = \frac{\widetilde{\alpha_{goal}^h} - \alpha_0}{p} < 1 + \frac{\widetilde{\alpha_{goal}} - \alpha_0}{p} \qquad \Longrightarrow \qquad n^* = \left\lceil (\widetilde{\alpha_{goal}} - \alpha_0)/p \right\rceil, \tag{E.7}$$

since n^* must be an integer. By Definition 2.4.3, $N = n^*$ for $\alpha \leq \widetilde{\alpha_{goal}}$. Thus, we get

$$N = n^* = \left\lceil (\widetilde{\alpha_{goal}} - \alpha_0)/p \right\rceil . \tag{E.8}$$

Since $\widetilde{\alpha_{goal}}$ corresponds to $Q_{goal} = Q_{max}(\widetilde{\alpha_{goal}})$, we can solve Eq. (A.2) for $\widetilde{\alpha_{goal}}$ by letting $Q_{max} = Q_{goal}$ and $\alpha\left(\frac{\Delta}{1-\beta}\right) = \mu$, which leads to Eq. (2.24). When Q_{goal} is small, i.e., $\widetilde{\alpha_{goal}}$ is small, the lower-bound function $\tau\sqrt{\alpha} = \sqrt{C_{max}} - \sqrt{2Q_h}$ given in Theorem 2.4.2 is tight, we can use

$$Q_{max}(\alpha, \tau) \approx (\tau\sqrt{\alpha} + \sqrt{2Q_h})^2 \tag{E.9}$$

to estimate Q_{max} as discussed in **(2)** (about Claim 2) of **Remarks on Theorem 2.4.2.** Substituting α, τ, and $Q_{max}(\alpha,\tau)$ by $\widetilde{\alpha_{goal}}$, $\widetilde{\tau}$, and Q_{goal} in Eq. (E.9), respectively, yields Eq. (2.25). Hence, the proof follows. ■

APPENDIX F

PROOF OF THEOREM 2.5.2

Proof. For the convenience of presentation, we first prove the Claim 2, and then we give the proof of the Claim 1.

Claim 2: Since $Q_{goal}(\alpha, \tau)$ is defined only for rate-gain parameter $\alpha > 0$, from Eq. (2.17) we can take the right-hand limit by letting α approach 0 from the right-hand side of $\alpha = 0$ as follows:

$$\lim_{\alpha\downarrow 0} Q_{goal}(\alpha,\tau) = \lim_{\alpha\downarrow 0}\left\{\frac{\alpha\Delta}{1-\beta}\left(T_{max} + \frac{\mu}{\alpha}\log\frac{\mu}{R_{max}}\right)\right\} + \lim_{\alpha\downarrow 0}\left\{\frac{\alpha}{2}T_{max}^2\right\}. \tag{F.1}$$

Plugging Eq. (2.14) into the second term of Eq (F.1) leads to

$$\lim_{\alpha\downarrow 0}\left\{\frac{\alpha}{2}T_{max}^2\right\} = \lim_{\alpha\downarrow 0}\left\{\frac{\alpha}{2}\left[\tau + \sqrt{\frac{2Q_h}{\alpha}}\right]^2\right\} = Q_h. \tag{F.2}$$

Applying the constraint of $\beta = 1 - \frac{\alpha}{\mu}\Delta$ (see the footnote of Theorem 2.4.1), and plugging Eqs. (2.13) and (2.14) into the first term of Eq. (F.1), we obtain

$$\begin{aligned}
&\lim_{\alpha\downarrow 0}\left\{\frac{\alpha\Delta}{1-\beta}\left(T_{max} + \frac{\mu}{\alpha}\log\frac{\mu}{R_{max}}\right)\right\}\\
&= \lim_{\alpha\downarrow 0}\left\{\frac{\mu\left(\tau\alpha + \sqrt{2Q_h\alpha}\right) + \mu^2\log\frac{\mu}{\mu+\left(\tau\alpha+\sqrt{2Q_h\alpha}\right)}}{\alpha}\right\} &\text{(F.3)}\\
&= \lim_{\alpha\downarrow 0}\left\{\frac{\mu\left(\tau\sqrt{\alpha} + \sqrt{\frac{Q_h}{2}}\right)\left(\tau\sqrt{\alpha} + \sqrt{2Q_h}\right)}{\mu + \left(\tau\alpha + \sqrt{2Q_h\alpha}\right)}\right\} &\text{(F.4)}\\
&= Q_h. &\text{(F.5)}
\end{aligned}$$

where Eq. (F.4) is obtained by taking the partial derivatives with respect to α over both the numerator and denominator of the fraction in Eq. (F.3). Plugging Eqs. (F.5) and (F.2) into Eq. (F.1), Eq. (2.44) follows, which proves Claim 2.

Claim 1: According to Lemma G.1.1 which is given and proved below in Appendix G, $Q_{goal}(\alpha, \tau)$ is a strictly increasing function with respect to α for $\tau > 0$ and $\alpha > 0$. On the other hand, from the proof of Claim 2, we already proved that

$$\lim_{\alpha \downarrow 0} Q_{goal}(\alpha, \tau) = 2Q_h. \tag{F.6}$$

which implies that the Claim 1, because α_n resulted by the α-control law defined in Eq. (2.5) only assumes the discrete values of $\alpha = \alpha_n$ for $n = 1, 2, \cdots, \infty$, and $\alpha = \alpha_n > 0$ must always hold. Thus, we obtain

$$\inf_{\tau>0,\, \alpha_n>0,\, n=1,2,\cdots,\infty} \{Q_{goal}(\alpha_n, \tau)\} = 2Q_h; \tag{F.7}$$

which completes the proof. ■

APPENDIX G

PROOF OF LEMMA G.1.1

G.1 Maximum Queue-Length Monotonicity Lemma

The following lemma has been verified and used on many occasions through both numerical solutions and simulation results. The lemma also plays a critical role in our previous analysis and derivations. While this lemma is relatively intuitive, it deserves a rigorous proof, which turns out to be not trivial as shown below.

Lemma G.1.1 *The maximum-queue-length function $Q_{goal}(\alpha, \tau)$ defined by Eq. (2.17) is a strictly monotonic-increasing function with respect to α, $\forall \tau > 0$ and $\forall \alpha > 0$.* ■

G.2 Proof of Maximum Queue-Length Monotonicity Lemma (Lemma G.1.1)

Proof. Since $Q_{goal}(\alpha, \tau)$ is defined only for $\tau > 0$ and $\alpha > 0$, where $Q_{goal}(\alpha, \tau)$ is continuous and differentiable, and thus, we can take the partial derivative of $Q_{goal}(\alpha, \tau)$

with respect to α as follows:

$$\begin{aligned}\frac{\partial Q_{goal}(\alpha,\tau)}{\partial\alpha} &= \frac{1}{2}\tau^2+\tau\sqrt{\frac{2Q_h}{\alpha}}-\tau\sqrt{\frac{Q_h}{2\alpha}}-\mu\sqrt{\frac{Q_h}{2}}\alpha^{-\frac{3}{2}} \\ &\quad +\frac{\mu^2}{\alpha^2}\log\left[1+\frac{\alpha}{\mu}\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)\right] \\ &\quad -\frac{\mu^2\left(\tau+\sqrt{\frac{Q_h}{2\alpha}}\right)}{\mu\alpha+\alpha^2\left(\tau+\sqrt{\frac{2Q_h}{\alpha}}\right)}. \end{aligned} \tag{G.1}$$

Again, we use the constraint of $\beta=1-\frac{\alpha}{\mu}\Delta$ (see the footnote of Theorem 2.4.1), in derivations of $\frac{\partial Q_{goal}(\alpha,\tau)}{\partial\alpha}$ in Eq. (G.1). Using Eq. (D.3), we define a new real-valued function $\kappa(\alpha,\tau)$ as follows:

$$\begin{aligned}\kappa(\alpha,\tau) &\triangleq \left(\frac{\alpha^2}{\mu^2}\right)\frac{\partial Q_{goal}(\alpha,\tau)}{\partial\alpha} \\ &= \frac{1}{2}\left(\frac{\tau\alpha}{\mu}\right)^2+\frac{\tau}{\mu^2}\sqrt{2Q_h}\alpha^{\frac{3}{2}}-\frac{\tau}{\mu^2}\sqrt{\frac{Q_h}{2}}\alpha^{\frac{3}{2}} \\ &\quad -\frac{1}{\mu}\sqrt{\frac{Q_h\alpha}{2}}+\log\frac{\mu+\tau\alpha+\sqrt{2Q_h\alpha}}{\mu} \\ &\quad -\frac{\left(\tau\alpha+\sqrt{\frac{Q_h\alpha}{2}}\right)}{\mu+(\tau\alpha+\sqrt{2Q_h\alpha})} \end{aligned} \tag{G.2}$$

Taking the partial derivative with respect to α on both sides of Eq. (G.2), we obtain

$$\begin{aligned}\frac{\partial\kappa(\alpha,\tau)}{\partial\alpha} &= \frac{1}{(\mu+\sqrt{2Q_h\alpha}+\tau\alpha)^2}\left[\frac{Q_h^{\frac{3}{2}}}{\mu}\sqrt{\frac{\alpha}{2}}+\frac{4\tau Q_h}{\mu}\alpha+\alpha^{\frac{3}{2}}\right. \\ &\quad \cdot\left(\frac{2\tau^2}{\mu}\sqrt{2Q_h}+\frac{3\tau^3}{\mu}\sqrt{\frac{Q_h}{2}}+\frac{3\tau Q_h^{\frac{3}{2}}}{\sqrt{2}\mu^2}+\frac{\tau^2}{2\mu}\sqrt{\frac{Q_h}{2}}\right) \\ &\quad +\alpha^2\left(\frac{2\tau^3}{\mu}+\frac{5\tau^2 Q_h}{\mu^2}\right)+\alpha^{\frac{5}{2}} \\ &\quad \left.\cdot\left(\frac{2\tau^3}{\mu^2}\sqrt{2Q_h}+\frac{3\tau^3}{2\mu^2}\sqrt{\frac{Q_h}{2}}\right)+\frac{\tau^4}{\mu^2}\alpha^3\right]>0. \end{aligned} \tag{G.3}$$

That is, Eq. (G.3) proves the following:

$$\frac{\partial\kappa(\alpha,\tau)}{\partial\alpha}>0, \qquad \forall\tau>0 \text{ and } \forall\alpha>0, \tag{G.4}$$

which implies that $\kappa(\alpha, \tau)$ is a strictly monotonic-increasing function with respect to α, $\forall \tau > 0$ and $\alpha > 0$. On the other hand, notice that

$$\lim_{\alpha \downarrow 0} \kappa(\alpha, \tau) = 0. \tag{G.5}$$

Combining Eq. (G.4) and Eq. (G.5), it follows that $\kappa(\alpha, \tau) > 0$, $\forall \alpha > 0$, and thus

$$\kappa(\alpha, \tau) = \left(\frac{\alpha^2}{\mu^2}\right) \frac{\partial Q_{goal}(\alpha, \tau)}{\partial \alpha} > 0, \quad \forall \tau > 0 \text{ and } \forall \alpha > 0 \tag{G.6}$$

Reducing Eq. (G.6), we get

$$\frac{\partial Q_{goal}(\alpha, \tau)}{\partial \alpha} > 0, \quad \forall \tau > 0 \text{ and } \forall \alpha > 0, \tag{G.7}$$

which completes the proof of Lemma G.1.1. ∎

APPENDIX H

PROOF OF THEOREM 2.5.3

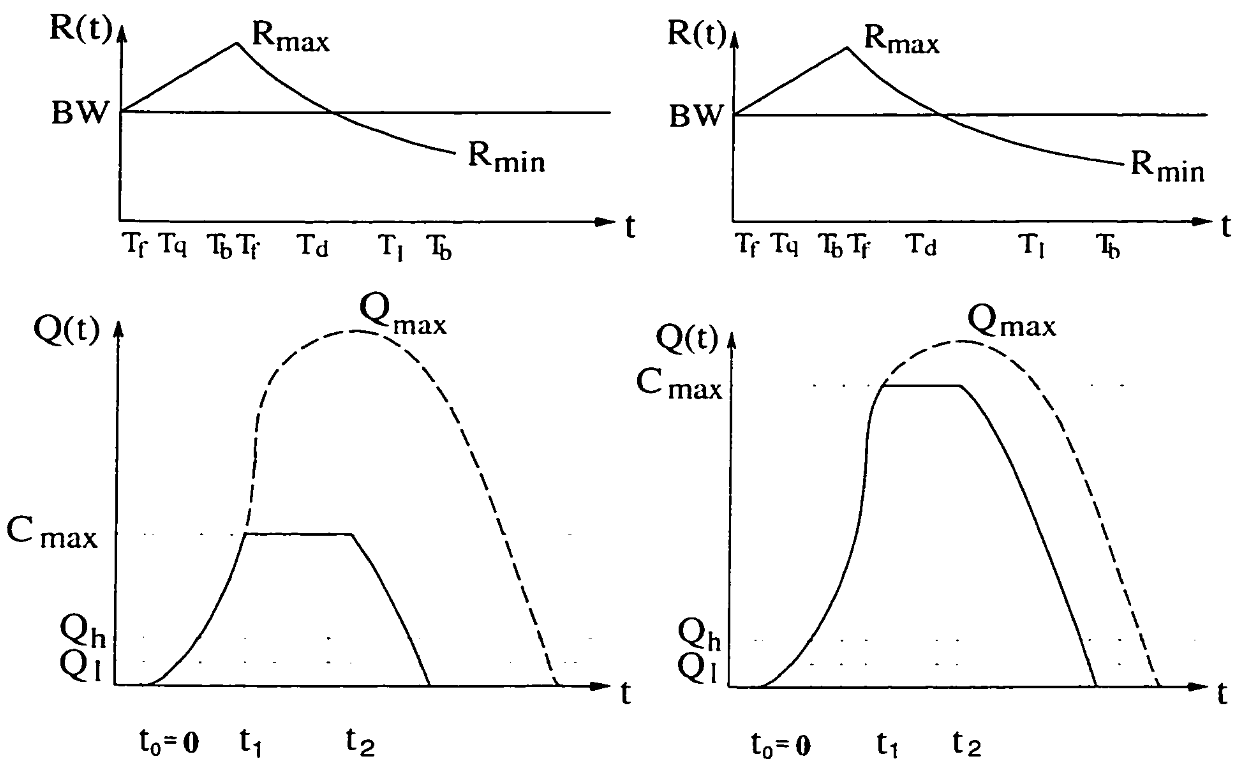
Proof. We first need to determine the upper and lower bounds of the "loss period" $[t_1, t_2]$, as shown in Figures H.1(a)–(b) (where the dotted lines of $Q(t)$ represent queue length if $Q_{max} < C_{max}$), within a rate-control cycle where t_1 (t_2) is the time when the router starts (stops) dropping packets. So, t_1 can be obtained by solving the bottleneck queue state equation (2.3) as:

$$Q(t_1) = \int_0^{t_1} [R(v - T_f) - \mu]\, dv = \xi = C_{max} \tag{H.1}$$

where, to simplify the calculations, the time-zero point is shifted to $t_0 = 0$ when $R(t - T_f)$ reaches bandwidth capacity μ = BW. Depending on the the rate-control parameters, t_1 can be either smaller (Figure H.1(a)), or larger (Figure H.1(b)), than $T_{max} \triangleq T_q + T_b + T_f$. Since T_{max} is the last moment $R(t)$ applies linear control, and t_1 is the time when $Q(t)$ hits $\xi = C_{max}$ for the first time, the conditions $t_1 \leq T_{max}$ and $t_1 > T_{max}$ can be equivalently expressed as $\xi \leq \frac{1}{2}\alpha T_{max}^2$ and $\xi > \frac{1}{2}\alpha T_{max}^2$, respectively. (From now on, we will use t_ξ to represent the lower bound of $[t_1, t_2]$.) These conditions generate the following two different cases in calculating $t_\xi \triangleq t_1$.

Case 1. If $\xi \leq \frac{1}{2}\alpha T_{max}^2$: $Q(t)$ is determined only by $R(t)$'s linear-control period, thus (see Figure H.1(a))

$$Q(t_\xi) = \int_0^{t_\xi} \alpha\, t\, dt = \xi \quad \Longrightarrow \quad t_\xi = \sqrt{\frac{2\xi}{\alpha}} \tag{H.2}$$



(a) Derivation of ρ — **Case 1.** (b) Derivation of ρ — **Case 2.**

Figure H.1: Derivation of number of lost packets ρ

which gives the first case of computing t_ξ in Theorem 2.5.3.

Case 2. If $\xi > \frac{1}{2}\alpha T_{max}^2$: $Q(t_\xi)$ is determined by both $R(t)$'s linear-control and exponential-control periods, thus (see Figure H.1(b))

$$Q(t_\xi) = \int_0^{T_{max}} \alpha\, t\, dt + \int_0^{t_\xi - T_{max}} \left(R_{max} e^{-(1-\beta)\frac{t}{\Delta}} - \mu\right) dt = \xi. \tag{H.3}$$

Eq. (2.47) follows by simplifying Eq. (H.3).

By definition of $[t_1, t_2]$, $R(t) \geq \mu$ must hold during $[t_1, t_2]$, which is the condition for $Q(t) = \xi$. During $[t_1, t_2]$, when $R(t) > \mu$, $Q(t)$ tends to increase, but upper-bounded by ξ, thus $Q(t) = \xi$. When $R(t) = \mu$, $Q(t) = \xi$ is maintained because the bottleneck queue length remains unchanged when the arrival rate at the bottleneck equals its departuring rate. Therefore, the upper bound t_2 is the time when $R(t)$ drops back exactly to μ and any further decrease in $R(t)$ will lead to $Q(t) < \xi$. By the definition of T_d in Eq. (2.16), we obtain $t_2 = T_{max} + T_d$.

During $[t_1, t_2]$, $R(t) \geq \mu$, and thus $R(t)$ can be written as: $R(t) = \mu + (R(t) - \mu)$. The first term μ maintains $Q(t) = \xi$, and the second term $(R(t) - \mu)$ causes packet drops. Therefore, the *packet-drop rate* is $(R(t) - \mu)$. Then, the number ρ of lost packets within one rate-control cycle, can be obtained by

$$\rho = \int_{t_1}^{t_2} [R(t) - \mu]\, dt = \int_{t_\xi}^{T_{max}+T_d} [R(t) - \mu]\, dt. \tag{H.4}$$

which is also divided into the following two cases, because $R(t)$ has different expressions, depending on $\xi \leq \frac{1}{2}\alpha T_{max}^2$ or $\xi > \frac{1}{2}\alpha T_{max}^2$.

Case 1. If $\xi \leq \frac{1}{2}\alpha T_{max}^2$: $R(t)$ consists of two parts, and thus (see Figure H.1(a))

$$\begin{aligned} \rho &= \int_{t_\xi}^{T_{max}+T_d} [R(t) - \mu]\, dt \\ &= \int_{t_\xi}^{T_{max}} \alpha t dt + \int_0^{T_d} \left(R_{max} e^{-(1-\beta)\frac{t}{\Delta}} - \mu \right) dt. \end{aligned} \tag{H.5}$$

Reducing Eq. (H.5) yields the first part of Eq. (2.46).

Case 2. If $\xi > \frac{1}{2}\alpha T_{max}^2$: $R(t)$ has only one part, and thus (see Figure H.1(b))

$$\begin{aligned} \rho &= \int_{t_\xi}^{T_{max}+T_d} [R(t) - \mu]\, dt = \int_0^{T_d} \left(R_{max} e^{-(1-\beta)\frac{t}{\Delta}} - \mu \right) dt \\ &\quad - \int_0^{t_\xi - T_{max}} \left(R_{max} e^{-(1-\beta)\frac{t}{\Delta}} - \mu \right) dt. \end{aligned} \tag{H.6}$$

Simplifying Eq. (H.6) leads to the second part of Eq. (2.46). This completes the poof. ■

APPENDIX I

PROOF OF THEOREM 2.6.1

Proof. We prove this theorem by considering transient and state and equilibrium states, respectively.

(I) In Transient State. The linear α-control function can be expressed by

$$\alpha_i(k+1) = \begin{cases} p_I + q_I\alpha_i(k); & \text{if } BCN(k-1,k) = (0,0), \\ p_D + q_D\alpha_i(k); & \text{if } BCN(k) = 1. \end{cases} \tag{I.1}$$

We now derive the constraints to determine the coefficients p_I, q_I, p_D, and q_D to guarantee convergence to both efficiency and fairness.

(1) Convergence to Efficiency. To ensure $\alpha_t(k)$ converges to its target α_{goal}, the α-control must be a negative feedback during each α-control cycle, i.e.,

$$\begin{cases} \alpha_t(k+1) > \alpha_t(k); & \text{if } BCN(k-1,k) = (0,0) \\ \alpha_t(k+1) < \alpha_t(k); & \text{if } BCN(k) = 1 \end{cases} \tag{I.2}$$

where $\alpha_t(k+1) = \sum_{i=1}^{n} \alpha_i(k+1)$ and $\alpha_t(k) = \sum_{i=1}^{n} \alpha_i(k)$. Using Eq. (I.1), we reduce Eq. (I.2) to

$$\begin{cases} q_I > 1 - \frac{np_I}{\sum_{i=1}^{n}\alpha_i(k)}, & \forall\ n \text{ and } \forall\ \sum_{i=1}^{n}\alpha_i(k); \quad \text{if } BCN(k-1,k) = (0,0), \\ q_D < 1 - \frac{np_D}{\sum_{i=1}^{n}\alpha_i(k)}, & \forall\ n \text{ and } \forall\ \sum_{i=1}^{n}\alpha_i(k); \quad \text{if } BCN(k) = 1, \end{cases} \tag{I.3}$$

(2) Convergence to Fairness. Convergence of $\alpha(k)$ to fairness can be expressed by

$$\lim_{k\to\infty} \phi(\alpha(k)) = \lim_{k\to\infty} \frac{[\sum_{i=1}^{n}\alpha_i(k)]^2}{n\ \sum_{i=1}^{n}\alpha_i^2(k)} = 1. \tag{I.4}$$

Plugging the linear-control function $g(\cdot,\cdot)$ into the fairness index given in *Definition 3* and defining $\theta \triangleq \frac{p}{q}$, we get

$$\begin{aligned}
\phi(\alpha(k+1)) &\triangleq \frac{[\sum_{i=1}^{n}\alpha_i(k+1)]^2}{n\sum_{i=1}^{n}\alpha_i^2(k+1)} = \frac{(\sum_{i=1}^{n}[p+q\alpha_i(k)])^2}{n\sum_{i=1}^{n}[p+q\alpha_i(k)]^2} \\
&= \frac{(\sum_{i=1}^{n}[\theta+\alpha_i(k)])^2}{n\sum_{i=1}^{n}[\theta+\alpha_i(k)]^2} = \phi(\alpha(k)) + [1-\phi(\alpha(k))]\left[1-\frac{\sum_{i=1}^{n}\alpha_i^2(k)}{\sum_{i=1}^{n}[\theta+\alpha_i(k)]^2}\right],
\end{aligned}$$

and further,

$$\phi(\alpha(k+1)) - \phi(\alpha(k)) = [1-\phi(\alpha(k))]\left[1-\frac{\sum_{i=1}^{n}\alpha_i^2(k)}{\sum_{i=1}^{n}[\theta+\alpha_i(k)]^2}\right] \tag{I.5}$$

Note that $\phi(\alpha(k+1)) - \phi(\alpha(k))$ in Eq. (I.5) is a monotonic-increasing function of $\theta \triangleq \frac{p}{q}$, and $\phi(\alpha(k+1)) \geq \phi(\alpha(k))$ *iff* $\theta \geq 0$. Thus, if $\theta > 0$, fairness increases: $\phi(\alpha(k+1)) > \phi(\alpha(k))$; if $\theta = 0$, the fairness remains unchanged: $\phi(\alpha(k+1)) = \phi(\alpha(k))$. Since $\theta \triangleq \frac{p_I}{q_I}$ in α-increase phase and $\theta \triangleq \frac{p_D}{q_D}$ in α-decrease phase, we get four possible cases as follows:

$$\begin{cases}
\textbf{1. } \text{if } \frac{p_D}{q_D} > 0 \wedge \frac{p_I}{q_I} > 0, & \text{then } \phi(\alpha(k+1)) > \phi(\alpha(k)) \text{ in} \\
& \text{both } \alpha\text{-decrease and } \alpha\text{-increase;} \\
\textbf{2. } \text{if } \frac{p_D}{q_D} > 0 \wedge \frac{p_I}{q_I} = 0, & \text{then } \phi(\alpha(k+1)) > \phi(\alpha(k)) \text{ in} \\
& \alpha\text{-decrease and } \phi(\alpha(k+1)) = \\
& \phi(\alpha(k)) \text{ in } \alpha\text{-increase;} \\
\textbf{3. } \text{if } \frac{p_D}{q_D} = 0 \wedge \frac{p_I}{q_I} > 0, & \text{then } \phi(\alpha(k+1)) = \phi(\alpha(k)) \text{ in} \\
& \alpha\text{-decrease and } \phi(\alpha(k+1)) > \\
& \phi(\alpha(k)) \text{ in } \alpha\text{-increase;} \\
\textbf{4. } \text{if } \frac{p_D}{q_D} = 0 \wedge \frac{p_I}{q_I} = 0, & \text{then } \phi(\alpha(k+1)) = \phi(\alpha(k)) \text{ in} \\
& \text{both } \alpha\text{-decrease and } \alpha\text{-increase;}
\end{cases} \tag{I.6}$$

Eq. (I.6) implies that control-function coefficients p_D, p_I, q_D, and q_I must all have the same sign if they are not zero. Combining Eq. (I.6) with Eq. (I.1), we conclude that these four control-function coefficients must be all positive if not zero, and q_I and q_D must be positive since $\alpha_i(k)$ $\forall$ i are always positive numbers. The convergence condition given in

Eq. (I.3) adds further constraints on q_D such that $0 < q_D < 1$. Thus, the constraints on the control-function coefficients, in terms of convergence to fairness and efficiency, can be summarized as

$$\textbf{constraint:}\{0 < q_D < 1, \quad 0 < q_I, \quad (p_D \geq 0 \wedge p_I > 0) \vee (p_D > 0 \wedge p_I \geq 0)\} \tag{I.7}$$

which includes cases **1**, **2**, and **3** as described in Eq. (I.6).

Since the α-control is exercised on a per-connection basis, and the i-th source does not have any information on $\alpha_j(k)$, $\forall\, j \neq i$ and value of n (α-control is a distributed algorithm), the convergence condition given in Eq. (I.3) cannot be explicitly used to further specify the control-function coefficients. In the absence of such information, each connection must satisfy the negative feedback condition as follows, which represents a stronger condition for convergence to efficiency:

$$\begin{cases} \alpha_i(k+1) > \alpha_i(k) \Longrightarrow p_I + (q_I - 1)\alpha_i(k) > 0, \forall\, i, & \text{if } BCN(k-1,k) = (0,0); \\ \alpha_i(k+1) < \alpha_i(k) \Longrightarrow p_D + (q_D - 1)\alpha_i(k) < 0, \forall\, i, & \text{if } BCN(k) = 1; \end{cases} \tag{I.8}$$

Eq. (I.8) yields further constraints in determining control-function coefficients. Since $(q_D - 1)\alpha_i(k)$ (< 0 due to Eq. (I.7)) may have an arbitrarily small absolute value, the second inequality in Eq. (I.8) requires $p_D = 0$, to fairness in α-increase. The first inequality in Eq. (I.8) requires $q_I \geq 1$ to ensure $p_I + (q_I - 1)\alpha_i(k) > 0\ \forall\, \alpha_i(k) > 0$. Since $\theta = \frac{p_I}{q_I}$ (fairness increases only in α-increase phase) and $\phi(\alpha(k+1)) - \phi(\alpha(k))$ is an increasing function of θ, we let q_I take its minimum $q_I = 1$ which is the *optimal* value for the convergence to fairness. Thus, we obtain the feasible and optimal linear control function defined by the following constraint:

$$\textbf{constraint:}\{0 < q_D < 1, q_I = 1, p_D = 0, p_I > 0\} \tag{I.9}$$

which is the exactly what we proposed for the α-control in the transient state, i.e.,

$$\alpha_i(k+1) = \begin{cases} p + \alpha_i(k); & \text{if } BCN(k-1,k) = (0,0) \\ q\alpha(k); & \text{if } BCN(k) = 1 \end{cases} \tag{I.10}$$

where $p_I = p > 0$, $q_I = 1$, $p_D = 0$, and $0 < q_D = q < 1$.

(II) In Equilibrium State. The linear α-control function is expressed by

$$\alpha_i(k+1) = \begin{cases} \frac{1}{q}\alpha_i(k); & \text{if } BCN(k-1,k) = (1,0), \\ q\alpha_i(k); & \text{if } BCN(k) = 1, \end{cases} \tag{I.11}$$

Since $p_D = 0$, $p_I = 0$, $q_D = q$ $(0 < q < 1)$, and $q_I = \frac{1}{q} > 1$, this control function belongs to case 4 in Eq. (I.6) where $\theta = 0$. Thus, the fairness is maintained as the α-control enters the equilibrium state. On the other hand, when $p_D = 0$ and $p_I = 0$, the constraints for convergence to efficiency become:

$$\textbf{constraint:}\{0 < q_D < 1, q_I > 1\} \tag{I.12}$$

which also satisfies Eqs. (I.8) and (I.3). Thus, the convergence to efficiency is also maintained for that connection. This completes the proof. ■

APPENDIX J

PROOF OF THEOREM 3.4.1

Proof. P_j's length is $j+1$ (in number of hops) and its leaf is located at the $(j+1)$-th level of the multicast tree (see Figure 3.2 for the case of $m=4$). Plugging Eq. (3.2) into Eq. (3.1), we get

$$\tau_u(j,\Delta) = \begin{cases} 2 + j\ \Delta, & \text{if } 2 \le \Delta \le \tau_{max} = 2m; \\ 2(j+1), & \text{if } \Delta = 1. \end{cases} \tag{J.1}$$

So, it suffices to prove Eq. (J.1), which consists of two parts to be proved as follows.

Part 1: Assume $2 \le \Delta \le \tau_{max} = 2m$. We can rewrite the first part of Eq. (J.1) as $\tau_u(j,\Delta) = 2 + j\Delta = (j+2) + j(\Delta-2) + j \triangleq C_1 + C_2 + C_3$. Then, the three components of $\tau_u(j,\Delta) = C_1 + C_2 + C_3$ constitute the RM-cell RTT under HBH as follows.

$C_1 = j+2$ is the time for a forward RM cell to traverse from the root to P_j's leaf node, then to return to the first branch node from the leaf toward the root (see Figure 3.2 for the case of $m=4$). It takes $(j+1)$ time units for a forward RM cell to reach P_j's leaf from the root and 1 time unit to immediately move one hop back to the first consolidating branch node, so $C_1 = (j+1)+1 = j+2$.

$C_2 = j(\Delta-2)$ is contributed by feedback RM cells waiting at branch nodes for the subsequent forward RM cells, each moving one hop upward the feedback RM cell at each branch node. Let's start with $\Delta = 2$, implying that feedback RM cells do not have to wait for forward RM cells in order to move upward as they arrive at each branch node at the same time as a forward RM cell arrives at the branch node. Thus, $C_2 = j(\Delta - 2) = 0$ holds. Now, suppose $\Delta = 2 + \ell$ with $\ell \geq 1$. Then, feedback RM cells always arrive at branch nodes ℓ time units earlier than forward RM cells, and thus, have to wait $\ell = (\Delta - 2)$ time units before making one hop move. So, $C_2 = j\,\ell = j\,(\Delta - 2)$, since there are j branch nodes along P_j.

$C_3 = j$ is the time for P_j's feedback RM cells to traverse from its first branch node from the leaf back to the root without waiting for forward RM cells to arrive, which is j, since there are j hops between the first branch node and the root.

Part 2: Assume $\Delta = 1$, then feedback RM cells will never wait for forward RM cells at any branch-node, implying that $C_2 = 0$, and $C_1 = j + 2$ and $C_3 = j$ remain the same as in Part 1. Thus, $\tau_u(j, \Delta) = C_1 + C_3 = 2(j + 1)$ for $\Delta = 1$, which gives the second part of Eq. (J.1). This completes the proof. ∎

APPENDIX K

PROOF OF LEMMA 3.4.1

Proof. When the switch algorithm checks for feedback RM-cell synchronization, that is the operation of ($conn\_patt\_vec \ominus resp\_branch\_vec = \underline{1}$) at a branch-node, the feedback RM cell on a shorter path P_i always arrives at the branch-node earlier than that (in response to the same forward RM cell) from a longer path $P_{\bar{i}}$. Thus, $P_{\bar{i}}$'s feedback RM cell can be synchronized, without waiting, *at least* with the feedback RM cell in response to the same forward RM cell, from the shorter path P_i at each branch-node. So, the feedback RM cell on a longer path never waits for feedback RM cells on a shorter path for feedback synchronization. ■

APPENDIX L

PROOF OF LEMMA 3.4.2

Proof. In contrast with the case described in Lemma 3.4.1, the feedback RM cell from a shorter path *may or may not* have to wait for the feedback RM cell from a longer path for feedback synchronization.

Claim 1 $\Longrightarrow$ Claim 2: If P_j's feedback RM cell does not wait at the first branch-node from P_j's leaf, then it becomes part of the feedback RM cell from a longer path after its RM cell is consolidated at the first branch-node. By Lemma 3.4.1, it does not wait for feedback RM cells from any path at all subsequent branch-nodes on P_j to achieve feedback synchronization.

Claim 2 $\Longrightarrow$ Claim 3: If P_j's feedback RM cell does not wait for a longer path's feedback at any branch-node, then at least it doesn't wait for synchronization at the first branch-node from P_j's leaf, i.e., P_j's feedback RM cell arrives at its first branch-node at the exact same time when the feedback RM cell from the longest path arrives at this branch-node. But it takes $(2m - j)$ time units for an RM cell to traverse from the root to the leaf of the longest path then return to P_j's first branch-node. On the other hand, it takes $(j + 2)$ time units for an RM cell to traverse from the root to P_j's leaf then return to its first branch-node. Thus, the arrival time of P_j's feedback RM cell at its first branch-node is $(j+2)+k\Delta$ where $k = 0, 1, \cdots$, corresponding to the $(k + 1)$-th RM cell, respectively. Then, $\exists\, k \in \{0, 1, \cdots\}$ such that the feedback RM cell from the longest path is synchronized with P_j's feedback

RM cell whose arrival time is $(j+2)+k\Delta$ at its first branch-node, and satisfies the following constraint (on k, for $1 \le j \le m-1$ and $1 \le \Delta \le \tau_{max} = 2m$):

$$(j+2) + k\Delta \le 2m - j < (j+2) + (k+1)\Delta. \tag{L.1}$$

But P_j's feedback RM cell does not wait for the feedback RM cell from a longer path at any branch-node on P_j. Thus, $\exists\, k \in \{0, 1, \cdots\}$ such that $(j+2)+k\Delta = 2m - j$, and hence Claim 3 follows.

Claim 3 $\Longrightarrow$ Claim 4: From $2(m-j-1) - k\Delta = 0$, we know that P_j's feedback RM cell does not wait for a longer path's feedback at the first branch-node from the leaf. According to the proof of Claim 1 $\Longrightarrow$ Claim 2, P_j's feedback RM cell does not wait for a longer path's feedback at all branch-nodes on P_j. Therefore, P_j's steady-state RM-cell roundtrip delay only consists of the pure transmission (propagation plus processing, but no waiting) delay, meaning that $\tau_u(j, \Delta) = 2(j+1)$. Since P_j's feedback RM cell may or may not have to wait for the feedback from a longer path for synchronization, depending on the value of Δ for given m and j, $\tau_u(j, \Delta)$ is lower-bounded by $2(j+1)$, and thus Claim 4 follows.

Claim 4 $\Longrightarrow$ Claim 1: If $\tau_u(j, \Delta)$ attains its minimum, then P_j's feedback RM cell does not wait for feedback RM cells from a longer path for synchronization at all branch-nodes on P_j, and hence not at the first branch-node from the leaf of P_j. This completes the proof. ∎

APPENDIX M

PROOF OF THEOREM 3.4.2

Proof. For convenience of presentation, we begin with Claim 2's proof.

Claim 2: By Claim 3 and Claim 4 of Lemma 3.4.2 and its proofs of Claim 2 $\Longrightarrow$ Claim 3 and Claim 3 $\Longrightarrow$ Claim 4, P_j's steady-state RM cell roundtrip delay $\tau_u(j, \Delta)$ can be expressed as the sum of transmission delay $2(j+1)$ and synchronization delay W_j

$$\tau_u(j, \Delta) = 2(j+1) + W_j \tag{M.1}$$

where W_j is the net waiting time for P_j's feedback RM cell to synchronize with the feedback on a longer path at the first branch-node from P_j's leaf. Based on Lemma 3.4.2's proof of Claim 2 $\Rightarrow$ Claim 3 and Eq. (L.1), $\exists\, k \in \{0, 1, 2, \cdots\}$ such that W_j can be expressed as

$$W_j = (2m - j) - [(j+2) + k\Delta] = 2(m - j - 1) - k\Delta. \tag{M.2}$$

Since the feedback RM cell on a longer path is always synchronized with the most recently arrived feedback on a shorter path at P_j's first branch-node as a constraint Eq. (L.1), the minimum possible synchronization-waiting time (≥ 0) determines W_j. By Lemma 3.4.1, $W_j \geq 0$. Thus, k in Eq. (M.2) is determined by

$$k_j^* \triangleq \max_{k \in \{0,1,2,\cdots\}} \{k \mid 2(m - j - 1) - k\Delta \geq 0\} \tag{M.3}$$

where $1 \leq j \leq m-1$ and k_j^* is obtained by minimizing $W_j = 2(m - j - 1) - k\Delta \geq 0$ over k. Combining Eqs. (M.1), (M.2), and (M.3), we get Eq. (3.6).

Claim 1: Let n_i be the number of P_j's feedback RM cells going through the initial state. Since the first feedback RM cell received by the root always experiences the longest path's roundtrip delay, in initial state the RM-cell roundtrip delay decreases from $\tau_{max} = 2m$ to its steady-state value $\tau_u(j, \Delta)$ ($\leq \tau_{max} = 2m$). Thus, the number of RM cells which go through the initial state is given by

$$\begin{aligned} n_i &= \frac{\tau_{max} - \tau_u(j,\Delta)}{\Delta} = \frac{\tau_{max} - (\tau_{max} - k_j^*\Delta)}{\Delta} = k_j^* \\ &= \max_{k \in \{0,1,2,\cdots\}} \{k \mid 2(m-j-1) - k\Delta \geq 0\} \end{aligned} \tag{M.4}$$

which results in Eq. (3.5).

Claim 3: Based on the proposed switch algorithm, all forward RM cells in the initial state are consolidated in the first feedback RM cell and sent back to the root at time $t = \tau_{max} = 2m$ to start feedback synchronization. In addition, the very first RM cell's roundtrip delay is always equal to $\tau_{max} = 2m$ for all paths. Thus, if $k_j^* \geq 1$ for P_j, the i-th $(1 \leq i \leq k_j^*)$ initial-state RM cell will experience a roundtrip delay of $\tau_u(j, \Delta, i) = \tau_{max} - (i-1)\Delta$, since it enters the system $(i-1)\Delta$ time units later than the very first RM cell. After k^* RM cells pass through the flow-controlled system (i.e., $i > k_j^*$ for P_j), the system reaches steady state and P_j's RM cell roundtrip delay becomes a constant (independent of i) specified by $\tau_u(j, \Delta, i) = \tau_u(j, \Delta)$. If $k_j^* = 0$ for P_j, i.e., P_j's feedback must be synchronized with those feedback RM cells corresponding to the same forward RM cell. Thus, the system enters steady state from the very first RM cell since P_j's RM cell roundtrip delay does not have initial-state (i.e., $k_j^* = 0$). Therefore, $\tau_u(j, \Delta, i) = 2m = \tau_{max}$, if $k_j^* = 0$ for P_j. This completes the proof. ■

APPENDIX N

PROOF OF THEOREM 3.5.1

Proof. Claim 1 $\Longrightarrow$ Claim 2: If $k_j^* = 0$, then $\tau(j, \Delta) = \tau_{max} = 2m$ according to Theorem 3.4.2. Thus, corresponding to the same forward RM cell, the feedback RM cell returned via path P_j arrives at the root node at the same time as the feedback RM cell returned via any longer path $P_{\bar{j}}$ $(1 \leq j < \bar{j} \leq m - 1)$. This implies that at the first consolidating branch-node from P_j's leaf, P_j's feedback RM cell is only synchronized with the feedback RM cells in response to the same forward RM cell, thus making P_j strictly-synchronized.
Claim 2 $\Longrightarrow$ Claim 3: If P_j is strictly-synchronized, then at the first consolidating branching-node from P_j's leaf, P_j's feedback RM cell is only synchronized with the feedback RM cells in response to the same forward RM cell. This implies that $\tau_u(j, \Delta) = \tau_u(\bar{j}, \Delta)$ $\forall \bar{j}$ such that $1 \leq j < \bar{j} \leq m - 1$, which includes the feedback RM cell from the longest path $P_{\bar{j}} = P_{m-1}$. By letting $\bar{j} = m-1$ in Eq. (3.6), of Claim 2 in Theorem 3.4.2, we get $\tau_u(j, \Delta) = \tau_{max} = 2m$.
Claim 3 $\Longrightarrow$ Claim 1: If $\tau_u(j, \Delta) = \tau_{max} = 2m$, then by Eq. (3.6) in Theorem 3.4.2, we get $k_j^* = 0$ since $\Delta \geq 1$. ■

APPENDIX O

PROOF OF THEOREM 3.5.2

Proof. Claim 1: The the equality part of Eq. (3.8) follows directly from Eq. (M.2), Eq. (M.3) in the proof of Theorem 3.4.2, and Lemma 3.4.1. We now prove the inequality part: $W_j < \Delta$, by contradiction. Assume $W_j \geq \Delta$. Then, by the equality part of Eq. (3.8), $W_j = 2(m-j-1) - k_j^* \Delta \geq \Delta$. Thus, we get $2(m-j-1) - (k_j^*+1)\Delta \geq 0$, which contradicts the definition of k_j^* given in Eq. (M.3).

Claim 2: If P_j is strictly-synchronized, then $k_j^* = 0$. Letting $k_j^* = 0$ in Eq. (3.8), we get $W_j = 2(m-j-1)$. But since $1 \leq j < m-1$ and $m > 2$, the desired result $W_j = 2(m-j-1) > 0$ follows.

Claim 3: We prove the sufficient condition first, and then the necessary condition.

"$\Longrightarrow$": If $W_j = 0$, then Eq. (3.8) is reduced to $2(m-j-1) = k_j^* \Delta$, i.e., $k_j^* = \frac{2(m-j-1)}{\Delta}$. Thus, $2(m-j-1)$ *mod* $\Delta = 0$.

"$\Longleftarrow$": If $2(m-j-1)$ *mod* $\Delta = 0$, then $\exists\, k \in \{0,1,2,\cdots\}$ such that $\frac{2(m-j-1)}{\Delta} = k$. But $k_j^* = \max_{k \in \{0,1,2,\cdots\}} \{k \mid 2(m-j-1) - k\Delta \geq 0\} = \lfloor \frac{2(m-j-1)}{\Delta} \rfloor = \frac{2(m-j-1)}{\Delta} = k$. Thus, we get $k = k_j^*$ and $2(m-j-1) = k_j^* \Delta$ which, by Eq. (3.8), implies $W_j = 2(m-j-1) - k_j^* \Delta = 0$. This completes the proof. ■

APPENDIX P

PROOF OF THEOREM 3.5.3

Proof. The proof of $\mathcal{P} = \mathcal{P}_S \ominus \mathcal{P}_N \ominus \mathcal{P}_W$ is trivial from Definition 3.5.1 and Definition 3.5.2. Now, we prove the three claims as follows.

Claim 1: Let j^* be the largest possible path number (from left to right) such that P_{j^*} is still not yet strictly-synchronized, i.e., $P_1, P_2, P_3, \cdots, P_{j^*}$ are either wait-free synchronized or non strictly-synchronized and $P_{j^*+1}, P_{j^*+2}, P_{j^*+3}, \cdots, P_{m-2}$ are all strictly-synchronized paths. Thus, by Definition 3.5.1 and Theorem 3.5.1, we have

$$k^*_{j^*} = \left\lfloor \frac{2(m - j^* - 1)}{\Delta} \right\rfloor \geq 1 \tag{P.1}$$

which reduces to

$$2(m - j^* - 1) \geq \Delta \quad \Longrightarrow \quad j^* \leq \frac{1}{2}[2(m-1) - \Delta]. \tag{P.2}$$

Since j^* is the largest integer that satisfies Eq. (P.1) or Eq. (P.2), we get $j^* = \left\lfloor \frac{1}{2}[2(m-1) - \Delta] \right\rfloor$. But the total number of paths under consideration is $(m-2)$ (P_{m-1} is a special case since it always has the same property as P_m, which is a trivial case) for a multicast tree of height m. Thus, $S_\Delta = (m-2) - j^* = (m-2) - \left\lfloor \frac{1}{2}[2(m-1) - \Delta] \right\rfloor = (m-2) - \left\lfloor (m-1) - \frac{\Delta}{2} \right\rfloor = (m-2) - \{(m-1) - \lceil \frac{\Delta}{2} \rceil\} = \lceil \frac{\Delta}{2} \rceil - 1$.

Claim 2: According to the sufficient and necessary condition to be a wait-free synchronized path, which is given in Claim 3 of Theorem 3.5.2, we want to determine the number of paths which satisfy $2(m - j - 1) \; mod \; \Delta = 0$ for $1 \leq j \leq m - 2$ and a given Δ. Let

$\mathcal{P}' \triangleq \{2(m-j-1) \mid j = 1, 2, 3, \cdots, (m-3), (m-2)\} = \{2(m-2), 2(m-3), 2(m-4), \cdots, 2\}$. Thus, $\mathcal{P}'$ defines a one-to-one mapping between elements of $\mathcal{P}'$ and all $(m-2)$ candidate paths, such that $2(m-2) \leftrightarrow P_1$, $2(m-3) \leftrightarrow P_2$, $2(m-4) \leftrightarrow P_3$, $\cdots$, $2 \leftrightarrow P_{m-2}$. Note that $\mathcal{P}'$ contains $(m-2)$ consecutive even numbers starting from 2. Therefore, we consider the following two cases.

Case 1: $\Delta =$ even. The number of wait-free synchronized paths, N_Δ, is determined by the elements in $\mathcal{P}'$ which is an integer multiple of Δ. Since the even numbers in $\mathcal{P}'$ are consecutive, we get

$$\begin{aligned} N_\Delta &\triangleq \|\{j \mid 2(m-j-1) \ mod \ \Delta = 0\}\| && \text{(P.3)} \\ &= \max_{N \in \{0,1,2,\cdots\}} \{N \mid 2(m-2) - N\Delta \geq 0\} && \text{(P.4)} \\ &= \left\lfloor \frac{2(m-2)}{\Delta} \right\rfloor && \text{(P.5)} \end{aligned}$$

where $1 \leq j \leq (m-2)$.

Case 2: $\Delta =$ odd. Since Δ is odd, only those elements of $\mathcal{P}'$, which contain both factors 2 and Δ, satisfy $2(m-j-1) \ mod \ \Delta = 0$. Thus, only those elements of $\mathcal{P}'$, which are integer multiples of (2Δ), contribute to N_Δ. Therefore, we get

$$\begin{aligned} N_\Delta &\triangleq \|\{j \mid 2(m-j-1) \ mod \ \Delta = 0\}\| && \text{(P.6)} \\ &= \max_{N \in \{0,1,2,\cdots\}} \{N \mid 2(m-2) - N(2\Delta) \geq 0\} && \text{(P.7)} \\ &= \left\lfloor \frac{(m-2)}{\Delta} \right\rfloor && \text{(P.8)} \end{aligned}$$

where $1 \leq j \leq (m-2)$. Combining Eqs. (P.5) and (P.8), Eq. (3.9) follows.

<u>Claim 3</u>: Applying <u>Claim 1</u> and <u>Claim 2</u> to fact $\mathcal{P} = \mathcal{P}_S \ominus \mathcal{P}_N \oplus \mathcal{P}_W$, Eq. (3.10) follows. This completes the proof. ∎

APPENDIX Q

PROOF OF THEOREM 4.4.1

Proof. For convenience of presentation, we start with the Claim 2.

Claim 2: An unbalanced multicast tree of height m, as defined as in Definition 4.3.1, consists of a set of $2m-1$ links, $\mathcal{L} = \{L_1, L_2, \cdots, L_{2m-1}\}$, which are labeled in the way as shown in Figure 4.1(a). Since $0 < p_i < 1$, it is possible that all these $2m-1$ links are not marked, and thus there is no dominant bottleneck path in the tree. On the other hand, if at least one of these $2m-1$ links is marked as the bottleneck link, then by Definition 4.3.2 the shortest path which contains the marked link(s) is the dominant bottleneck path. According to the structure defined by Definitions 4.3.1 and 4.3.2, the dominant bottleneck path is unique. Thus, there is at most one dominant bottleneck path.

In what follows, we use $L_i = 1$ (0) to represent that link L_i is (not) marked as the bottleneck. Thus, $\Pr\{L_i = 1\} = p_i$ and $\Pr\{L_i = 0\} = 1 - p_i$. By Definitions 4.3.1 and 4.3.2, the probability that P_1 becomes the dominant bottleneck path is equal to the probability that $L_1 = 1$ or $L_2 = 1$, implying

$$\begin{aligned} \psi(P_1, m) &= \Pr\{L_1 = 1 \cup L_2 = 1\} = 1 - \Pr\{L_1 = 0 \cap L_2 = 0\} \\ &= 1 - \Pr\{L_1 = 0\}\Pr\{L_2 = 0\} \qquad \text{(Q.1)} \\ &= 1 - (1-p_1)(1-p_2) = p_1 + p_2 - p_1 p_2. \qquad \text{(Q.2)} \end{aligned}$$

where Eq. (Q.1) is due to **C3** of Definition 4.3.1. Thus, the first part of Eq. (4.5) follows from Eq. (Q.2).

Consider P_k, $1 < k \leq m-1$. Since the last two links are L_{2k-1} and L_{2k} (see Figure 4.1(a)), the probability that P_k becomes the dominant bottleneck path is equal to the probability that $L_i = 0$, $\forall i \in \{1, 2, \cdots, 2(k-1)\}$ and $L_{2k-1} = 1$ or $L_{2k} = 1$, which leads to

$$\begin{aligned}
\psi(P_k, m) &= \Pr\left\{\bigcap_{i=1}^{2(k-1)} L_i = 0 \cap \{L_{2k-1}=1 \cup L_{2k}=1\}\right\} \\
&= \Pr\left\{\bigcap_{i=1}^{2(k-1)} L_i = 0\right\} \Pr\{L_{2k-1}=1 \cup L_{2k}=1\} \\
&= (p_{2k-1} + p_{2k} - p_{2k-1}p_{2k}) \prod_{i=1}^{2(k-1)} (1 - p_i) \qquad \text{(Q.3)}
\end{aligned}$$

where the second and third equalities of Eq. (Q.3) are due to **C3** of Definition 4.3.1 and the proof of Eq. (Q.2). Thus, the second part of Eq. (4.5) follows from Eq. (Q.3).

For the last path P_m, since the last link is L_{2m-1}, the probability that P_m becomes the dominant bottleneck path is equal to the probability that $L_i = 0$, $\forall i \in \{1, 2, \cdots, 2(m-1)\}$ and $L_{2m-1} = 1$, which leads to

$$\psi(P_m, m) = \Pr\left\{\bigcap_{i=1}^{2(m-1)} L_i = 0 \cap \{L_{2m-1} = 1\}\right\} = p_{2m-1} \prod_{i=1}^{2(m-1)} (1 - p_i). \qquad \text{(Q.4)}$$

where the second equality of Eq. (Q.4) is due to **C3** of Definition 4.3.1. Hence, the third part of Eq. (4.5) follows from Eq. (Q.4).

Claim 1: The proof of Eq. (4.3) follows from the proof of the second part of Eq. (4.5). Now, we prove that the probability mass function defined by Eq. (4.3) satisfies the following normalization condition:

$$
\begin{aligned}
&\lim_{m\to\infty}\sum_{k=1}^{m}\psi(P_k,\infty)\\
&=\lim_{m\to\infty}\sum_{k=1}^{m}\left\{\left[1-(1-p_{2k-1})(1-p_{2k})\right](1-p_1)(1-p_2)(1-p_3)\cdots(1-p_{2k-3})(1-p_{2k-2})\right\}\\
&=\lim_{m\to\infty}\left\{\left[1-(1-p_1)(1-p_2)\right]+\left[1-(1-p_3)(1-p_4)\right](1-p_1)(1-p_2)\right.\\
&\qquad\qquad+\left[1-(1-p_5)(1-p_6)\right](1-p_1)(1-p_2)(1-p_3)(1-p_4)+\cdots\\
&\qquad\qquad\left.+\left[1-(1-p_{2m-1})(1-p_{2m})\right](1-p_1)(1-p_2)\cdots(1-p_{2m-3})(1-p_{2m-2})\right\}\\
&=\lim_{m\to\infty}\left\{1-(1-p_1)(1-p_2)\cdots(1-p_{2m-3})(1-p_{2m-2})(1-p_{2m-1})(1-p_{2m})\right\}\\
&=1-\lim_{m\to\infty}\prod_{i=1}^{2m}(1-p_i) \qquad\qquad (Q.5)
\end{aligned}
$$

But since the second limiting term of Eq. (Q.5) satisfies the following facts:

$$
0 \;\leq\; \lim_{m\to\infty}\prod_{i=1}^{2m}(1-p_i) \leq \lim_{m\to\infty}\prod_{i=1}^{2m}(1-p_{min}) = \lim_{m\to\infty}(1-p_{min})^{2m} = 0 \qquad (Q.6)
$$

where $0 < p_{min} \triangleq \min_{i\in\{1,2,\cdots,\infty\}}\{p_i\} < 1$, we obtain

$$
\lim_{m\to\infty}\sum_{k=1}^{m}\psi(P_k,\infty) = 1. \qquad (Q.7)
$$

Thus, $\psi(P_k,\infty)$, $\forall\, k \in \{1,2,\cdots,\infty\}$ defines a valid probability mass function. In addition, Eq. (Q.7) also implies that there exist at least one dominant bottleneck path as $m\to\infty$. On the other hand, according to the tree structure defined by Definitions 4.3.1 and 4.3.2, there is at most one dominant bottleneck path. Thus, there exists one and only one dominant bottleneck path, which completes the proof. ∎

APPENDIX R

PROOF OF THEOREM 4.4.2

Proof. Claim 1: It follows directly from Claim 2: of Theorem 4.4.1 by letting $p_i = p$.

Claim 2: Since $\psi(P_k, p, m)$ is a real-valued continuous function of p and differentiable for $0 < p < 1$, we can take a partial derivative of $\psi(P_k, p, m)$ with respect to p and set it to zero. For different ranges of k, we have the following two cases.

Case 1. $1 \leq k \leq m - 1$:

$$\frac{\partial \psi}{\partial p}(P_k, p, m) = (2-p)(1-p)^{2(k-1)} - p(1-p)^{2(k-1)} - 2(k-1)p(2-p)(1-p)^{2k-3} = 0. \quad \text{(R.1)}$$

Reducing Eq. (R.1), we get the following quadratic equation and its solutions:

$$kp^2 - 2kp + 1 = 0 \text{ which has two roots:} \Rightarrow p_1 = 1 + \sqrt{\frac{k-1}{k}} \text{ and } p_2 = 1 - \sqrt{\frac{k-1}{k}}. \quad \text{(R.2)}$$

Taking the meaningful solution from Eq. (R.2) to satisfy the probability constraint, $0 < p < 1$, we get

$$p^* = \arg \max_{0<p<1} \psi(P_k, p, m) = 1 - \sqrt{\frac{k-1}{k}} \quad \text{(R.3)}$$

which is unique and gives the first part of the Eq. (4.9).

Case 2. $k = m$:

$$\frac{\partial \psi}{\partial p}(P_m, p, m) = (1-p)^{2(m-1)} - 2(m-1)p(1-p)^{2m-3} = 0 \quad \text{(R.4)}$$

Reducing and solving Eq. (R.4) for p, we get the unique solution:

$$p^* = \arg \max_{0<p<1} \psi(P_m, p, m) = \frac{1}{2m-1} \quad \text{(R.5)}$$

which is the second part of Eq. (4.9).

Then, plugging Eqs. (R.3) and (R.5) into the first part and the second part of Eq. (4.7), respectively, Eq. (4.8) follows.

Claim 3: Eqs. (4.11) and (4.10) follow by plugging Eq. (4.7), and Theorem 3.4.1's Eq. (3.1) and Theorem 3.4.2's Eq. (3.6), respectively, into Eq. (R.6) that follows below:

$$\overline{\tau}(m) = E[\,\tau(m)\,] = \sum_{j=1}^{m} \tau_u(j,\Delta)\psi(P_j,p,m) \tag{R.6}$$

Claim 4: Eqs. (4.13) and (4.12) follow by plugging Eq. (4.7), and Theorem 3.4.1's Eq. (3.1) and Theorem 3.4.2's Eq. (3.6), respectively, into Eq. (R.7) that follows below:

$$\sigma^2(m) = Var[\,\tau(m)\,] \;=\; \sum_{j=1}^{m}[\tau_u(j,\Delta)]^2\psi(P_j,p,m) - \left(\sum_{j=1}^{m} \tau_u(j,\Delta)\psi(P_j,p,m)\right)^2 \tag{R.7}$$

This completes the proof. ∎

APPENDIX S

PROOF OF COROLLARY 4.4.1

Proof. Claim 1: Eq. (4.15) follows from Eq. (4.7) by letting $m \to \infty$ and observing that $p(1-p)^{2(m-1)} \to 0$ as $m \to \infty$. Eq (4.16) follows from the proof of Claim 1 of Theorem 4.4.1, which is also verified by the following direct proof:

$$\begin{aligned} \lim_{m\to\infty} \sum_{k=1}^{m} \psi(P_k, p, \infty) &= \lim_{m\to\infty} \sum_{k=1}^{m} p(2-p)(1-p)^{2(k-1)} = \sum_{k=1}^{\infty} p(2-p)(1-p)^{2(k-1)} \\ &= p(2-p) \sum_{k=1}^{\infty} \left[(1-p)^2\right]^{k-1} = \frac{p(2-p)}{1-(1-p)^2} = 1. \end{aligned} \tag{S.1}$$

Claim 2: The first part of Eq. (4.17) follows from the proof of Eq. (4.8). The second part of Eq. (4.17) holds because

$$\lim_{k\to\infty} \frac{1}{k}\left(1-\frac{1}{k}\right)^{k-1} = \lim_{k\to\infty} \frac{1}{k-1}\left(1-\frac{1}{k}\right)^{k} = 0 \tag{S.2}$$

where we notice $\lim_{k\to\infty} \left(1-\frac{1}{k}\right)^k = e^{-1}$. Eq. (4.18) follows from the proof of Eq. (4.9).

Claim 3: Eqs. (4.19) and (4.20) follow immediately from Eqs. (4.11) and (4.10) by letting $m \to \infty$.

Claim 4: Eqs. (4.21) and (4.22) are the immediate results from Eqs. (4.13) and (4.12) by letting $m \to \infty$. This completes the proof. ∎

APPENDIX T

PROOF OF THEOREM 5.2.1

Proof. For convenience of presentation, we start with Claim 2.

Claim 2: An unbalanced multicast tree of height m in Definition 5.2.1, consists of a set of $2m-1$ links, $\mathcal{L} = \{L_1, L_2', L_2, L_3', L_3, \cdots, L_m', L_m\}$, which are labeled as in Figure 5.1(a). Since $0 < p_i < 1$, it is possible that all of these $2m-1$ links are not marked, and hence, no dominant bottleneck path exists in the tree. On the other hand, if at least one of these $2m-1$ links is marked as the bottleneck link, then, by Definition 5.2.2, the shortest path which contains the marked link(s) is the dominant bottleneck. According to the structure defined by Definitions 5.2.1 and 5.2.2, the dominant bottleneck path is unique. Thus, there is at most one dominant bottleneck path.

By Definitions 5.2.1 and 5.2.2, the probability that P_1 becomes the dominant bottleneck path is equal to the probability that $X_1 = 1$ or $X_2' = 1$, which yields the first part of Eq. (5.12) as follows:

$$\begin{aligned} \psi_d(P_1, m) &= \Pr\{X_1 = 1 \cup X_2' = 1\} = 1 - \Pr\{X_1 = 0 \cap X_2' = 0\} \\ &= 1 - \Pr\{X_1 = 0\}\Pr\{X_2' = 0 \mid X_1 = 0\} \end{aligned} \tag{T.1}$$

Consider path P_k, $2 \leq k \leq m-1$. Since the last two links are L_k and L_{k+1}' (see Figure 5.1(a)), the probability that P_k becomes the dominant bottleneck path is equal to the probability that $X_i = 0$, $\forall i \in \{1, 2, \cdots, k-1\}$ and $X_i' = 0$, $\forall i \in \{2, 3, \cdots, k\}$, and

$X_k = 1$ or $X'_{k+1} = 1$, which leads to

$$
\begin{aligned}
\psi_d(P_k, m) &= \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\} \cap \{X_k = 1 \cup X'_{k+1} = 1\}\right\} \\
&= \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, X_k = 1\right\} \\
&\quad + \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, X'_{k+1} = 1\right\} \\
&\quad - \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, X_k = 1, X'_{k+1} = 1\right\} \\
&= \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, X_k = 1\right\} \\
&\quad - \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, X_k = 1, X'_{k+1} = 1\right\} \\
&\quad + \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, \{X_k = 0 \cup X_k = 1\}, X'_{k+1} = 1\right\} \\
&= \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, X_k = 1\right\} \\
&\quad + \Pr\left\{\bigcap_{i=1}^{k-1}\{X_i = 0, X'_{i+1} = 0\}, X_k = 0, X'_{k+1} = 1\right\} \\
&= \Pr\{X_1 = 0\}\Pr\{X_2 = 0 \mid X_1 = 0\}\Pr\{X_3 = 0 \mid X_2 = 0\}\cdots \\
&\quad \cdot\Pr\{X_{k-1} = 0 \mid X_{k-2} = 0\}\Pr\{X'_2 = 0 \mid X_1 = 0\} \\
&\quad \cdot\Pr\{X'_3 = 0 \mid X_2 = 0\}\cdots\Pr\{X'_k = 0 \mid X_{k-1} = 0\} \\
&\quad \cdot\Pr\{X_k = 1 \mid X_{k-1} = 0\} + \Pr\{X_1 = 0\}\Pr\{X_2 = 0 \mid X_1 = 0\} \\
&\quad \cdot\Pr\{X_3 = 0 \mid X_2 = 0\}\cdots\Pr\{X_k = 0 \mid X_{k-1} = 0\}\Pr\{X'_2 = 0 \mid X_1 = 0\} \\
&\quad \cdot\Pr\{X'_3 = 0 \mid X_2 = 0\}\cdots\Pr\{X'_k = 0 \mid X_{k-1} = 0\} \\
&\quad \cdot\Pr\{X'_{k+1} = 1 \mid X_k = 0\} \qquad (T.2) \\
&= \Pr\{X_1 = 0\}\Pr\{X'_k = 0 \mid X_{k-1} = 0\}\Big[\Pr\{X_k = 1 \mid X_{k-1} = 0\} \\
&\quad + \Pr\{X_k = 0 \mid X_{k-1} = 0\}\Pr\{X'_{k+1} = 1 \mid X_k = 0)\Big] \\
&\quad \cdot\prod_{i=1}^{k-2}\left\{\Pr\{X_{i+1} = 0 \mid X_i = 0\}\Pr\{X'_{i+1} = 0 \mid X_i = 0\}\right\}, \qquad (T.3)
\end{aligned}
$$

where Eq. (T.2) is due to **C3** and **C4** of Definition 5.2.1. Thus, the second part of Eq. (5.12) follows from Eq. (T.3).

The probability that P_m becomes the dominant bottleneck path is equal to the probability that $X_i = 0$, $\forall i \in \{1, 2, \cdots, m-1\}$ and $X'_i = 0$, $\forall i \in \{2, 3, \cdots, m\}$, and $X_m = 1$, which implies:

$$\begin{aligned}
\psi_d(P_k, m) &= \Pr\left\{\bigcap_{i=1}^{m-1}\{X_i = 0, X'_{i+1} = 0\}, X_m = 1\right\} \\
&= \Pr\{X_1 = 0\}\Pr\{X_m = 1 \mid X_{m-1} = 0\}\Pr\{X'_m = 0 \mid X_{m-1} = 0\} \\
&\quad \cdot \prod_{i=1}^{m-2}\left\{\Pr\{X_{i+1} = 0 \mid X_i = 0\}\Pr\{X'_{i+1} = 0 \mid X_i = 0\}\right\} \qquad \text{(T.4)}
\end{aligned}$$

where Eq. (T.4) follows from the proof of the first term of Eq. (T.2) and is also due to **C3** and **C4** of Definition 5.2.1. Hence, the third part of Eq. (5.12) follows from Eq. (T.4).

Claim 1: The proof of Eq. (5.10) follows from the proof of the first and second parts of Eq. (5.12). Now, we prove that the probability mass function defined by Eq. (5.10) satisfies the following normalization condition

$$
\begin{aligned}
&\lim_{m\to\infty}\sum_{k=1}^{m}\psi_d(P_k,\infty)\\
&= \lim_{m\to\infty}\Bigg\{1-\Pr\{X_1=0\}\Pr\{X_2'=0\mid X_1=0\}+\Pr\{X_1=0\}\\
&\qquad\cdot\Pr\{X_2'=0\mid X_1=0\}\Big[\Pr\{X_2=1\mid X_1=0\}+\Pr\{X_2=0\mid X_1=0\}\\
&\qquad\cdot\Pr\{X_3'=1\mid X_2=0\}\Big]+\Pr\{X_1=0\}\Big[\Pr\{X_2=0\mid X_1=0\}\\
&\qquad\cdot\Pr\{X_2'=0\mid X_1=0\}\Big]\Pr\{X_3'=0\mid X_2=0\}\Big[\Pr\{X_3=1\mid X_2=0\}\\
&\qquad+\Pr\{X_3=0\mid X_2=0\}\Pr\{X_4'=1\mid X_3=0\}\Big]+\cdots\\
&\qquad+\Pr\{X_1=0\}\Big[\Pr\{X_2=0\mid X_1=0\}\Pr\{X_2'=0\mid X_1=0\}\\
&\qquad\cdot\Pr\{X_3=0\mid X_2=0\}\Pr\{X_3'=0\mid X_2=0\}\Pr\{X_4=0\mid X_3=0\}\\
&\qquad\cdot\Pr\{X_4'=0\mid X_3=0\}\cdots\Pr\{X_{m-1}=0\mid X_{m-2}=0\}\\
&\qquad\cdot\Pr\{X_{m-1}'=0\mid X_{m-2}=0\}\Big]\Pr\{X_m'=0\mid X_{m-1}=0\}\\
&\qquad\cdot\Big[\Pr\{X_m=1\mid X_{m-1}=0\}+\Pr\{X_m=0\mid X_{m-1}=0\}\\
&\qquad\cdot\Pr\{X_{m+1}'=1\mid X_m=0\}\Big]\Bigg\}\\
&= \lim_{m\to\infty}\Bigg\{1-\Pr\{X_1=0\}\Pr\{X_2'=0\mid X_1=0\}+\Pr\{X_1=0\}\\
&\qquad\cdot\Pr\{X_2'=0\mid X_1=0\}\Big[\Big(1-\Pr\{X_2=0\mid X_1=0\}\Big)\\
&\qquad+\Pr\{X_2=0\mid X_1=0\}\Big(1-\Pr\{X_3'=0\mid X_2=0\}\Big)\Big]\\
&\qquad+\Pr\{X_1=0\}\Big[\Pr\{X_2=0\mid X_1=0\}\Pr\{X_2'=0\mid X_1=0\}\Big]\\
&\qquad\cdot\Pr\{X_3'=0\mid X_2=0\}\Big[\Big(1-\Pr\{X_3=0\mid X_2=0\}\Big)\\
&\qquad+\Pr\{X_3=0\mid X_2=0\}\Big(1-\Pr\{X_4'=0\mid X_3=0\}\Big)\Big]\\
&\qquad+\cdots
\end{aligned}
\tag{T.5}
$$

where Eq. (T.5) continues on the next page in Eq. (T.6) as follows.

The equation Eq. (T.6) that follows below continues from the Eq. (T.5) in the last page.

$$
\begin{aligned}
&+\cdots+\Pr\{X_1=0\}\Big[\Pr\{X_2=0\mid X_1=0\}\Pr\{X_2'=0\mid X_1=0\} \qquad \text{(T.6)}\\
&\cdot\Pr\{X_3=0\mid X_2=0\}\Pr\{X_3'=0\mid X_2=0\}\Pr\{X_4=0\mid X_3=0\}\\
&\cdot\Pr\{X_4'=0\mid X_3=0\}\cdots\Pr\{X_{m-1}=0\mid X_{m-2}=0\}\\
&\cdot\Pr\{X_{m-1}'=0\mid X_{m-2}=0\}\Big]\Pr\{X_m'=0\mid X_{m-1}=0\}\\
&\cdot\Big[\Big(1-\Pr\{X_m=0\mid X_{m-1}=0\}\Big)+\Pr\{X_m=0\mid X_{m-1}=0\}\\
&\cdot\Big(1-\Pr\{X_{m+1}'=0\mid X_m=0\}\Big)\Big]\Big\}\\
=\lim_{m\to\infty}&\Big\{1-\Pr\{X_1=0\}\Pr\{X_2'=0\mid X_1=0\}+\Pr\{X_1=0\}\\
&\cdot\Pr\{X_2'=0\mid X_1=0\}\Big[1-\Pr\{X_2=0\mid X_1=0\}\Pr\{X_3'=0\mid X_2=0\}\Big]\\
&+\Pr\{X_1=0\}\Big[\Pr\{X_2=0\mid X_1=0\}\\
&\cdot\Pr\{X_2'=0\mid X_1=0\}\Big]\Pr\{X_3'=0\mid X_2=0\}\Big[1-\Pr\{X_3=0\mid X_2=0\}\\
&\cdot\Pr\{X_4'=0\mid X_3=0\}\Big]+\cdots+\Pr\{X_1=0\}\Big[\Pr\{X_2=0\mid X_1=0\}\\
&\cdot\Pr\{X_2'=0\mid X_1=0\}\Pr\{X_3=0\mid X_2=0\}\Pr\{X_3'=0\mid X_2=0\}\\
&\cdot\Pr\{X_4=0\mid X_3=0\}\Pr\{X_4'=0\mid X_3=0\}\cdots\Pr\{X_{m-1}=0\mid X_{m-2}=0\}\\
&\cdot\Pr\{X_{m-1}'=0\mid X_{m-2}=0\}\Big]\Pr\{X_m'=0\mid X_{m-1}=0\}\\
&\cdot\Big[1-\Pr\{X_m=0\mid X_{m-1}=0\}\Pr\{X_{m+1}'=0\mid X_m=0\}\Big]\Big\}\\
=\lim_{m\to\infty}&\Big\{1-\Pr\{X_1=0\}\Big[\Pr\{X_2=0\mid X_1=0\}\Pr\{X_2'=0\mid X_1=0\}\\
&\cdot\Pr\{X_3=0\mid X_2=0\}\Pr\{X_3'=0\mid X_2=0\}\Pr\{X_4=0\mid X_3=0\}\\
&\cdot\Pr\{X_4'=0\mid X_3=0\}\cdots\Pr\{X_{m-1}=0\mid X_{m-2}=0\}\\
&\cdot\Pr\{X_{m-1}'=0\mid X_{m-2}=0\}\Big]\Pr\{X_m'=0\mid X_{m-1}=0\}\\
&\cdot\Pr\{X_m=0\mid X_{m-1}=0\}\Pr\{X_{m+1}'=0\mid X_m=0\}\Big\}\\
=\lim_{m\to\infty}&\Big\{1-\Pr\{X_1=0\}\Pr\{X_{m+1}'=0\mid X_m=0\}\\
&\cdot\prod_{i=1}^{m-1}\Big[\Pr\{X_{i+1}=0\mid X_i=0\}\Pr\{X_{i+1}'=0\mid X_i=0\}\Big]\Big\}
\end{aligned}
$$

$$= 1 - \Pr\{X_1 = 0\} \lim_{m\to\infty} \Pr\{X'_{m+1} = 0 \mid X_m = 0\}$$
$$\cdot \lim_{m\to\infty} \prod_{i=1}^{m-1} \left[\Pr\{X_{i+1} = 0 \mid X_i = 0\} \Pr\{X'_{i+1} = 0 \mid X_i = 0\} \right] \quad (T.7)$$

But since the limiting terms of Eq. (T.7) satisfy the following facts:

$$\begin{aligned} 0 &\leq \lim_{m\to\infty} \Pr\{X'_{m+1} = 0 \mid X_m = 0\} \\ &\quad \cdot \lim_{m\to\infty} \prod_{i=1}^{m-1} \left[\Pr\{X_{i+1} = 0 \mid X_i = 0\} \Pr\{X'_{i+1} = 0 \mid X_i = 0\} \right] \\ &\leq \lim_{k\to\infty} \prod_{i=1}^{k-1} \left[p_{max} p'_{max} \right] = \lim_{k\to\infty} \left[p_{max} p'_{max} \right]^{k-1} \\ &= 0 \end{aligned} \quad (T.8)$$

where without loss generality we assume there exist subsequences[1] such that $0 < \Pr\{X_{n_i+1} = 0 \mid X_{n_i} = 0\} < 1$ and $0 < \Pr\{X'_{m_i+1} = 0 \mid X_{m_i} = 0\} < 1$ for $n_1 < n_2 < n_3 < \cdots$ and $m_1 < m_2 < m_3 < \cdots$, and

$$\begin{cases} 0 < p_{max} \triangleq \max_{i \in \{n_1, n_2, \cdots, \infty\}} \left\{ \Pr\{X_{i+1} = 0 \mid X_i = 0\} \right\} < 1; \\ 0 < p'_{max} \triangleq \max_{i \in \{m_1, m_2, \cdots, \infty\}} \left\{ \Pr\{X'_{i+1} = 0 \mid X_i = 0\} \right\} < 1; \end{cases} \quad (T.9)$$

Thus, we obtain:

$$\lim_{m\to\infty} \sum_{k=1}^{m} \psi_d(P_k, \infty) = 1. \quad (T.10)$$

Thus, $\psi_d(P_k, \infty), \forall\, k \in \{1, 2, \cdots, \infty\}$ defines a valid probability mass function. In addition, Eq. (T.10) also implies that there exist at least one dominant bottleneck path as $m \to \infty$. On the other hand, based on the tree structure defined by Definitions 5.2.1 and 5.2.2, there is at most one dominant bottleneck path. Thus, there exists one and only one dominant bottleneck path, which completes the proof. ∎

[1] The case for $\Pr\{X_{i+1} = 0 \mid X_i = 0\} \equiv 1$ and $\Pr\{X'_{i+1} = 0 \mid X_i = 0\} \equiv 1$, $\forall i$, is trivial, and it is easy to prove the theorem still holds for the trivial case.

APPENDIX U

PROOF OF THEOREM 5.3.1

Proof. Claim 1: Consider link-marking states X_i and X_{i+1} over links L_i and L_{i+1} for $i = 1, 2, \cdots$. Using the partition rule, we can write the Markov chain $\{X_i\}$'s state probability (for $X_{i+1} = 0$) at link L_{i+1} as follows:

$$\begin{aligned}\Pr\{X_{i+1} = 0\} &= \Pr\{X_i = 0\}\Pr\{X_{i+1} = 0 \mid X_i = 0\} \\ &\quad + \Pr\{X_i = 1\}\Pr\{X_{i+1} = 0 \mid X_i = 1\}.\end{aligned} \tag{U.1}$$

By defining

$$w^{(i)} \triangleq \Pr\{X_{i+1} = 0 \mid X_i = 0\} \quad \text{and} \quad v^{(i)} \triangleq \Pr\{X_{i+1} = 0 \mid X_i = 1\}, \tag{U.2}$$

Eq. (U.1) reduces to

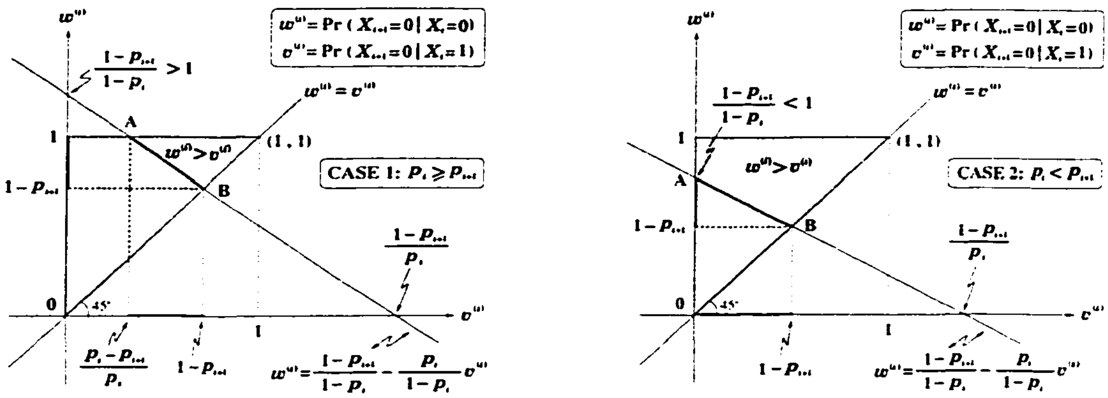
$$w^{(i)} = \frac{1 - p_{i+1}}{1 - p_i} - \frac{p_i}{1 - p_i} v^{(i)} \tag{U.3}$$

which defines a fundamental relationship between the two condition distributions $w^{(i)}$ and $v^{(i)}$ for the given marginal marking distributions p_i and p_{i+1} by the function $f(\cdot)$ as follows:

$$w^{(i)} \triangleq f(v^{(i)}) \triangleq \frac{1 - p_{i+1}}{1 - p_i} - \frac{p_i}{1 - p_i} v^{(i)}. \tag{U.4}$$

Our goal is to find a general system functional

$$w^{(i)} = \varphi(\alpha_i, p_i, p_{i+1}). \tag{U.5}$$



(a) **CASE 1:** $p_i \geq p_{i+1}$ (b) **CASE 2:** $p_i < p_{i+1}$

Figure U.1: Markov-chain dependency-degree modeling for **CASE 1** and **CASE 2**

which expresses the conditional distribution $w^{(i)}$ as a function of the Markov-chain dependency-degree factor $\alpha_i \in [0, 1]$, and the marginal probability distributions p_i and p_{i+1}.

Then we can solve for the upper and lower bounds for $w^{(i)} = \varphi(\alpha_i, p_i, p_{i+1})$ such that the following three constraints are satisfied:

C1. $(w^{(i)}, v^{(i)}) \in \left\{ (w^{(i)}, v^{(i)}) \mid w^{(i)} = f(v^{(i)}) \right\}$: where $f(\cdot)$ is defined in Eq. (U.4);

C2. $w^{(i)} > v^{(i)}$: because the Markov chain $\{X_i\}$ is positively dependent (see Definition 5.3.1);

C3. $0 \leq w^{(i)}, v^{(i)} \leq 1$: because $w^{(i)}, v^{(i)}$ are both probabilities.

We need to consider the following two cases, depending on $p_i \geq p_{i+1}$ or $p_i < p_{i+1}$.

CASE 1: $p_i \geq p_{i+1}$. To help present the proof, Figure U.1(a) plots the derived feasible solution regions, under the above three constraints **C1**, **C2**, and **C3**, for **CASE 1** in a 2-dimensional space spanned by $v^{(i)}$ and $w^{(i)}$ as the horizontal and vertical axis, respectively. **C1** states that all solution points must be on the straight line defined by $w^{(i)} = f(v^{(i)}) = \dfrac{1-p_{i+1}}{1-p_i} - \dfrac{p_i}{1-p_i} v^{(i)}$; **C2** says that all solution points must be within the region between the positive half axis of $w^{(i)}$ and the 45^0 straight line $w^{(i)} = v^{(i)}$ (the shaded area in Figure U.1(a)); **C3** requires that all solution points must be within the unit square area $w^{(i)} \in [0, 1]$ and $v^{(i)} \in [0, 1]$.

Applying **C1** through **C3**, the solution point set for $\{(w^{(i)}, v^{(i)})\}$ lie between points A and B along the straight line $w^{(i)} = f(v^{(i)}) = \dfrac{1-p_{i+1}}{1-p_i} - \dfrac{p_i}{1-p_i}v^{(i)}$. After some algebraic manipulation, we can show that the projection points of A and B onto $v^{(i)}$ and $w^{(i)}$ axises are $w_A^{(i)} = 1, w_B^{(i)} = 1 - p_{i+1}$ and $v_A^{(i)} = \dfrac{p_i - p_{i+1}}{p_i}, v_B^{(i)} = 1 - p_{i+1}$, respectively.

Then, the projection points of A and B onto the $w^{(i)}$ axis give $w^{(i)}$'s upper bounds $w_{max}^{(i)}$ and lower bound $w_{min}^{(i)}$, respectively. Likewise, the projection points of A and B onto the $v^{(i)}$ axis yields $v^{(i)}$'s lower bounds $v_{min}^{(i)}$ and upper bound $v_{max}^{(i)}$, respectively. That is,

$$\left\{\begin{array}{l} w_{min}^{(i)} = w_B^{(i)} = 1-p_{i+1}; \\ \\ w_{max}^{(i)} = w_A^{(i)} = 1; \end{array}\right. \implies 1-p_{i+1} \leq \Pr\{X_{i+1}=0 \,|\, X_i=0\} \leq 1, \qquad \text{(U.6)}$$

which proves the first part of Eq. (5.13). Similarly,

$$\left\{\begin{array}{l} v_{min}^{(i)} = v_A^{(i)} = \dfrac{p_i-p_{i+1}}{p_i}; \\ \\ v_{max}^{(i)} = v_B^{(i)} = 1-p_{i+1}; \end{array}\right. \implies \frac{p_i-p_{i+1}}{p_i} \leq \Pr\{X_{i+1}=0 \,|\, X_i=1\} \leq 1-p_{i+1}, \qquad \text{(U.7)}$$

which leads to the first part of Eq. (5.15).

CASE 2: $p_i < p_{i+1}$. Figure U.1(b) plots the derived feasible solution regions, under the three constraints **C1** through **C3**, for **CASE 2** in the same axis-coordinate system. Using constraints **C1** through **C3** we obtain similar results to those under **CASE 1**. However, the straight line $w^{(i)} = f(v^{(i)}) = \dfrac{1-p_{i+1}}{1-p_i} - \dfrac{p_i}{1-p_i}v^{(i)}$ intersects with the $w^{(i)}$ axis at point smaller than 1 while **CASE 1**'s corresponding intersection point is larger than 1, because $p_i < p_{i+1}$. This requires recalculation of the changed projection points of straight line between A and B onto $v^{(i)}$ and $w^{(i)}$ axises, which are shown to be $w_A^{(i)} = \dfrac{1-p_{i+1}}{1-p_i}, w_B^{(i)} = 1-p_{i+1}$ and $v_A^{(i)} = 0, v_B^{(i)} = 1-p_{i+1}$, respectively.

Thus, the new projection points of A and B onto the $w^{(i)}$ and $v^{(i)}$ axises yield **CASE 2**'s upper and lower bounds for $w^{(i)}$ and $v^{(i)}$, respectively. That is,

$$\left\{\begin{array}{l} w_{min}^{(i)} = w_B^{(i)} = 1-p_{i+1}; \\ \\ w_{max}^{(i)} = w_A^{(i)} = \dfrac{1-p_{i+1}}{1-p_i}; \end{array}\right. \implies 1-p_{i+1} \leq \Pr\{X_{i+1}=0 \,|\, X_i=0\} \leq \frac{1-p_{i+1}}{1-p_i}, \qquad \text{(U.8)}$$

which proves the second part of Eq. (5.13). Likewise,

$$\left\{\begin{array}{l} v_{min}^{(i)} = v_A^{(i)} = 0; \\ \\ v_{max}^{(i)} = v_B^{(i)} = 1 - p_{i+1}; \end{array}\right. \qquad \Longrightarrow \quad 0 \le \Pr\{X_{i+1} = 0 \,|\, X_i = 1\} \le 1 - p_{i+1}, \tag{U.9}$$

which proves the second part of Eq. (5.15).

Notice that adding both sides of Eq. (5.13) to those of Eq. (5.14), and Eq. (5.15) to those of Eq. (5.16), equals 1, respectively. We expected this result, because they are two mutually-complement events and must satisfy the normalization condition, i.e., $\Pr\{X_{i+1} = 0 \mid X_i = 0\} + \Pr\{X_{i+1} = 1 \mid X_i = 0\} = 1$ and $\Pr\{X_{i+1} = 0 \mid X_i = 1\} + \Pr\{X_{i+1} = 1 \mid X_i = 1\} = 1$. However, we need to prove that this still holds under the proposed Markov-chain dependency-degree model by independently proving Eqs. (5.14) and (5.16). For this, we apply the partition rule again over the state probability of $X_{i+1} = 1$:

$$\begin{aligned} \Pr\{X_{i+1} = 1\} \quad = \quad & \Pr\{X_i = 0\}\Pr\{X_{i+1} = 1 \mid X_i = 0\} \\ & + \Pr\{X_i = 1\}\Pr\{X_{i+1} = 1 \mid X_i = 1\}. \end{aligned} \tag{U.10}$$

By defining

$$\overline{w}^{(i)} \triangleq \Pr\{X_{i+1} = 1 \mid X_i = 0\} \quad \text{and} \quad \overline{v}^{(i)} \triangleq \Pr\{X_{i+1} = 1 \mid X_i = 1\}, \tag{U.11}$$

Eq. (U.10) reduces to

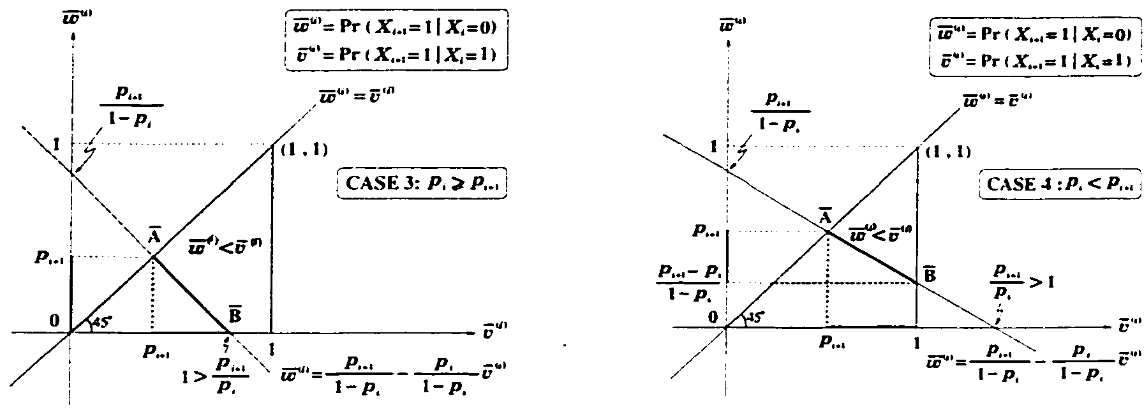
$$\overline{w}^{(i)} = \frac{p_{i+1}}{1 - p_i} - \frac{p_i}{1 - p_i}\overline{v}^{(i)} \tag{U.12}$$

Thus, the fundamental relationship between $\overline{w}^{(i)}$ and $\overline{v}^{(i)}$ is given by

$$\overline{w}^{(i)} \triangleq \overline{f}(\overline{v}^{(i)}) \triangleq \frac{p_{i+1}}{1 - p_i} - \frac{p_i}{1 - p_i}\overline{v}^{(i)}. \tag{U.13}$$

We can now solve for the upper and lower bounds for $\overline{w}^{(i)}$ subject to the following three constraints:

$\overline{C}$**1.** $(\overline{w}^{(i)}, \overline{v}^{(i)}) \in \left\{ (\overline{w}^{(i)}, \overline{v}^{(i)}) \mid \overline{w}^{(i)} = \overline{f}(\overline{v}^{(i)}) \right\}$: where $\overline{f}(\cdot)$ is defined in Eq. (U.13);



(a) **CASE 3:** $p_i \geq p_{i+1}$ (b) **CASE 4:** $p_i < p_{i+1}$

Figure U.2: Markov-chain dependency-degree modeling for **CASE 3** and **CASE 4**

$\overline{C}$**2.** $\overline{w}^{(i)} < \overline{v}^{(i)}$: because the Markov chain $\{X_i\}$ is positively dependent (see Definition 5.3.1);

$\overline{C}$**3.** $0 \leq \overline{w}^{(i)}, \overline{v}^{(i)} \leq 1$: because $\overline{w}^{(i)}, \overline{v}^{(i)}$ are probabilities.

Figure U.2 plots the derived feasible solution regions, under the three constraints $\overline{C}$**1** through $\overline{C}$**3** which also generate two different cases (**CASE 3** and **CASE 4**, depending on $p_i \geq p_{i+1}$ or $p_i < p_{i+1}$, respectively) as follows.

CASE 3: $p_i \geq p_{i+1}$. Figure U.2(a) plots the derived feasible solution regions, under the above three constraints $\overline{C}$**1**, $\overline{C}$**2**, and $\overline{C}$**3**, for **CASE 3** in a 2-dimensional space spanned by $\overline{v}^{(i)}$ and $\overline{w}^{(i)}$ as the horizontal and vertical axis, respectively. $\overline{C}$**1** states that all solution points must be on the straight line defined by $\overline{w}^{(i)} = f(\overline{v}^{(i)}) = \frac{p_{i+1}}{1-p_i} - \frac{p_i}{1-p_i}\overline{v}^{(i)}$; $\overline{C}$**2** says that all solution points must be within the region between the positive half axis of $\overline{v}^{(i)}$ and the 45^0 straight line $\overline{w}^{(i)} = \overline{v}^{(i)}$ (the shaded area in Figure U.2(a) — Notice that constraint $\overline{C}$**2** is opposite to **C2**, which makes the feasible solution area (shaded area in Figure U.2) flip down to the area between $\overline{v}^{(i)}$'s positive-half axis and 45^0 line); $\overline{C}$**3** requires that all solution points must be within the unit square area $\overline{w}^{(i)} \in [0, 1]$ and $\overline{v}^{(i)} \in [0, 1]$.

Applying $\overline{C}$**1** through $\overline{C}$**3**, the solution point set for $\{(\overline{w}^{(i)}, \overline{v}^{(i)})\}$ lie between points $\overline{A}$ and $\overline{B}$ along the straight line $\overline{w}^{(i)} = f(\overline{v}^{(i)}) = \frac{p_{i+1}}{1-p_i} - \frac{p_i}{1-p_i}\overline{v}^{(i)}$. After some algebraic manipulation, we can show that the projection points of $\overline{A}$ and $\overline{B}$ onto $\overline{v}^{(i)}$ and $\overline{w}^{(i)}$ axises are $\overline{w}^{(i)}_{\overline{A}} = p_{i+1}, \overline{w}^{(i)}_{\overline{B}} = 0$ and $\overline{v}^{(i)}_{\overline{A}} = p_{i+1}, \overline{v}^{(i)}_{\overline{B}} = \frac{p_{i+1}}{p_i}$, respectively.

Then, the projection points of $\overline{A}$ and $\overline{B}$ onto the $\overline{w}^{(i)}$ axis give $\overline{w}^{(i)}$'s upper bounds $\overline{w}^{(i)}_{max}$ and lower bound $\overline{w}^{(i)}_{min}$, respectively. Likewise, the projection points of $\overline{A}$ and $\overline{B}$ onto the $\overline{v}^{(i)}$ axis yields $\overline{v}^{(i)}$'s lower bounds $\overline{v}^{(i)}_{min}$ and upper bound $\overline{v}^{(i)}_{max}$, respectively. That is,

$$\begin{cases} \overline{w}^{(i)}_{min} = \overline{w}^{(i)}_{\overline{B}} = 0; \\ \\ \overline{w}^{(i)}_{max} = \overline{w}^{(i)}_{\overline{A}} = p_{i+1}; \end{cases} \Longrightarrow \quad 0 \leq \Pr\{X_{i+1}=1 \,|\, X_i=0\} \leq p_{i+1}, \tag{U.14}$$

which proves the first part of Eq. (5.14). Similarly,

$$\begin{cases} \overline{v}^{(i)}_{min} = \overline{v}^{(i)}_{\overline{A}} = p_{i+1}; \\ \\ \overline{v}^{(i)}_{max} = \overline{v}^{(i)}_{\overline{B}} = \dfrac{p_{i+1}}{p_i}; \end{cases} \Longrightarrow \quad p_{i+1} \leq \Pr\{X_{i+1}=1 \,|\, X_i=1\} \leq \frac{p_{i+1}}{p_i}, \tag{U.15}$$

which leads to the first part of Eq. (5.16).

CASE 4: $p_i < p_{i+1}$. Figure U.2(b) plots the derived feasible solution regions, under $\overline{C}$**1** through $\overline{C}$**3** for **CASE 4** in the same axis-coordinate system. Using $\overline{C}$**1**, $\overline{C}$**2**, and $\overline{C}$**3**, we obtain similar results to those under **CASE 3**. However, the straight line $\overline{w}^{(i)} = f(\overline{v}^{(i)}) = \frac{p_{i+1}}{1-p_i} - \frac{p_i}{1-p_i}\overline{v}^{(i)}$ intersects the $\overline{v}^{(i)}$ axis at point larger than 1 while **CASE 3**'s corresponding intersection point is smaller than 1, because $p_i < p_{i+1}$. This needs to recalculate the changed projection points of straight line between $\overline{A}$ and $\overline{B}$ onto $\overline{v}^{(i)}$ and $\overline{w}^{(i)}$ axises, which are shown to be $\overline{w}^{(i)}_{\overline{A}} = p_i, \overline{w}^{(i)}_{\overline{B}} = \frac{p_{i+1} - p_i}{1-p_i}$ and $\overline{v}^{(i)}_{\overline{A}} = p_{i+1}, \overline{v}^{(i)}_{\overline{B}} = 1$, respectively.

Thus, the new projection points of $\overline{A}$ and $\overline{B}$ onto the $\overline{w}^{(i)}$ and $\overline{v}^{(i)}$ axises yield **CASE 4**'s the upper and lower bounds for $\overline{w}^{(i)}$ and $\overline{v}^{(i)}$, respectively. That is,

$$\begin{cases} \overline{w}^{(i)}_{min} = \overline{w}^{(i)}_{\overline{B}} = \dfrac{p_{i+1} - p_i}{1-p_i}; \\ \\ \overline{w}^{(i)}_{max} = \overline{w}^{(i)}_{A} = p_{i+1}; \end{cases} \Longrightarrow \quad \frac{p_{i+1} - p_i}{1-p_i} \leq \Pr\{X_{i+1}=1 \,|\, X_i=0\} \leq p_{i+1}, \tag{U.16}$$

which proves the second part of Eq. (5.14). Likewise,

$$\begin{cases} \overline{v}^{(i)}_{min} = \overline{v}^{(i)}_{\overline{A}} = p_{i+1}; \\ \\ \overline{v}^{(i)}_{max} = \overline{v}^{(i)}_{\overline{B}} = 1; \end{cases} \implies p_{i+1} \leq \Pr\{X_{i+1}=1 \,|\, X_i=1\} \leq 1, \tag{U.17}$$

which proves the second part of Eq. (5.16).

<u>**Claim 2:**</u> Since we want to derive the function of

$$\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\} = \varphi(\alpha_i, p_i, p_{i+1}), \tag{U.18}$$

which can model all possible dependency-degrees between X_{i+1} and X_i, we introduce a real-valued Markov-chain dependency-degree factor $\alpha_i \in [0, 1]$ by which we define an exponential average between the upper and lower bounds of $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$ derived in <u>**Claim 1**</u> to evaluate the conditional probability distribution $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$. That is,

$$\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\} \triangleq$$
$$\begin{cases} w^{(i)} = \Pr\{X_{i+1}=0 \mid X_i=0\} = w^{(i)}_{min} + \alpha_i\left(w^{(i)}_{max} - w^{(i)}_{min}\right), & \text{if } x_{i+1}=0 \wedge x_i=0; \\ \overline{w}^{(i)} = \Pr\{X_{i+1}=1 \mid X_i=0\} = \overline{w}^{(i)}_{min} + (1-\alpha_i)\left(\overline{w}^{(i)}_{max} - \overline{w}^{(i)}_{min}\right), & \text{if } x_{i+1}=1 \wedge x_i=0; \\ v^{(i)} = \Pr\{X_{i+1}=0 \mid X_i=1\} = v^{(i)}_{min} + (1-\alpha_i)\left(v^{(i)}_{max} - v^{(i)}_{min}\right), & \text{if } x_{i+1}=0 \wedge x_i=1; \\ \overline{v}^{(i)} = \Pr\{X_{i+1}=1 \mid X_i=1\} = \overline{v}^{(i)}_{min} + \alpha_i\left(\overline{v}^{(i)}_{max} - \overline{v}^{(i)}_{min}\right), & \text{if } x_{i+1}=1 \wedge x_i=1; \end{cases} \tag{U.19}$$

where $w^{(i)}_{min}$, $w^{(i)}_{max}$, $v^{(i)}_{min}$, $v^{(i)}_{max}$, $\overline{w}^{(i)}_{min}$, $\overline{w}^{(i)}_{max}$, $\overline{v}^{(i)}_{min}$, and $\overline{v}^{(i)}_{max}$ are defined by Eqs. (U.6), (U.8), (U.7), (U.9), (U.14), (U.16), (U.15), and (U.17), respectively, depending on $p_i \geq p_{i+1}$ or $p_i < p_{i+1}$.

Notice that because $w^{(i)}$ and $\overline{w}^{(i)}$ are probabilities of two mutually-complement events, and so are $v^{(i)}$ and $\overline{v}^{(i)}$, in Eq. (U.19) we need to use two complementary dependency-degree factors α_i and $(1 - \alpha_i)$ to calculate the exponential average. According to the <u>**Claim 1**</u> proved above, the conditional distribution of $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$ must take a value between its upper and lower bounds. Thus, by tuning the Markov-chain dependency-degree factor α_i from 0 to 1, Eq. (U.19) ensures that $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$ can take any possible values between its upper and lower bounds. This proves that there exists a real-valued number $\alpha_i \in [0,1]$ such that all possible dependency-degrees between random variables X_i and X_{i+1} can be measured by the Markov-chain dependency-degree factor $\alpha_i \in [0,1]$.

Eq. (5.17) needs to be proved for all four different conditional distributions of $\Pr\{X_{i+1} = x_{i+1} \mid X_i = x_i\}$, respectively, each corresponding to one of the four different combinations of (x_i, x_{i+1}) where $x_i, x_{i+1} \in \{0,1\}$. Here we only give the proof for $\Pr\{X_{i+1} = 0 \mid X_i = 0\}$ and the proofs for the other three combination cases ($\Pr\{X_{i+1} = 1 \mid X_i = 0\}$, $\Pr\{X_{i+1} = 0 \mid X_i = 1\}$, and $\Pr\{X_{i+1} = 1 \mid X_i = 1\}$), which are omitted for lack of space, can be obtained in the way similar to the proof for $\Pr\{X_{i+1} = 0 \mid X_i = 0\}$ that follows below. To prove Eq. (5.17) for $x_i = x_{i+1} = 0$, we also need to consider the following two cases:

CASE I: $p_i \geq p_{i+1}$. Using Eqs. (U.6) and (U.19), we obtain:

$$w^{(i)} = \Pr\{X_{i+1} = 0 \mid X_i = 0\} = \begin{cases} 1 - p_{i+1}, & \text{if } \alpha_i = 0; \\ 1, & \text{if } \alpha_i = 1; \end{cases} \tag{U.20}$$

We prove two opposite directions of the *iff* condition as follows:

"$\Longrightarrow$":

$$\Pr\{X_{i+1} = 0, X_i = 0\} = \Pr\{X_i = 0\}\Pr\{X_{i+1} = 0 \mid X_i = 0\} = \Pr\{X_i = 0\}w^{(i)} \tag{U.21}$$

$$= \begin{cases} \Pr\{X_i = 0\}(1 - p_{i+1}), & \text{if } \alpha_i = 0; \\ \Pr\{X_i = 0\}, & \text{if } \alpha_i = 1; \end{cases} \tag{U.22}$$

$$= \begin{cases} \Pr\{X_i = 0\}\Pr\{X_{i+1} = 0\}, & \text{if } \alpha_i = 0; \\ \Pr\{X_i = 0\}, & \text{if } \alpha_i = 1; \end{cases} \tag{U.23}$$

The first part of Eq. (U.23) says if $\alpha_i = 0$, then $\Pr\{X_{i+1} = 0, X_i = 0\} = \Pr\{X_{i+1} = 0\}\Pr\{X_i = 0\}$, i.e., $\{X_i = 0\}$ and $\{X_{i+1} = 0\}$ are independent. The second part of Eq. (U.23) says if $\alpha_i = 1$, then $\Pr\{X_{i+1} = 0, X_i = 0\} = \Pr\{X_i = 0\}$, implying that $\{X_i = 0\}$ and $\{X_{i+1} = 0\}$ are "perfectly" dependent. This is because $\Pr\{X_{i+1} = 0, X_i = 0\} = \Pr\{X_i = 0\}$ if and only if $\{X_i = 0\}$ is a sub-event[1] of $\{X_{i+1} = 0\}$. Also, here $\{X_i = 0\}$

[1] The identical event is the special case of sub-event.

is a sub-event of $\{X_{i+1} = 0\}$ because $p_i \geq p_{i+1}$ implies $\Pr\{X_i = 0\} = 1 - p_i \leq 1 - p_{i+1} = \Pr\{X_{i+1} = 0\}$.

"$\Longleftarrow$": If $\{X_{i+1} = 0\}$ and $\{X_i = 0\}$ are independent, then $\Pr\{X_{i+1} = 0 \mid X_i = 0\} = \Pr\{X_{i+1} = 0\} = 1 - p_{i+1} = w^{(i)} \mid_{\alpha_i=0}$ for $p_i \geq p_{i+1}$, where the last equation ($1 - p_{i+1} = w^{(i)} \mid_{\alpha_i=0}$) is due to Eq. (U.20). Thus, we obtain: $\alpha_i = 0$, which proves the first part of Eq. (5.17). On the other hand, if $\{X_i = 0\}$ and $\{X_{i+1} = 0\}$ are perfectly dependent and because $p_i \geq p_{i+1}$, then $\{X_i = 0\}$ is a sub-event of $\{X_{i+1} = 0\}$ as shown in the above. This implies that if $\{X_i = 0\}$ occurs, then $\{X_{i+1} = 0\}$ must occur under the positive dependence given by Definition 5.3.1. This means that $\Pr\{X_{i+1} = 0 \mid X_i = 0\} = 1 = w^{(i)} \mid_{\alpha_i=1}$ for $p_i \geq p_{i+1}$, where the last equation ($1 = w^{(i)} \mid_{\alpha_i=1}$) is due to Eq. (U.20). Thus, we obtain: $\alpha_i = 1$, which proves the second part of Eq. (5.17).

CASE II: $p_i < p_{i+1}$. Using Eqs. (U.8) and (U.19), we obtain:

$$w^{(i)} = \Pr\{X_{i+1} = 0 \mid X_i = 0\} = \begin{cases} 1 - p_{i+1}, & \text{if } \alpha_i = 0; \\ \dfrac{1 - p_{i+1}}{1 - p_i}, & \text{if } \alpha_i = 1; \end{cases} \tag{U.24}$$

The independent parts ($\alpha_i = 0$) proof remain the same as in **CASE I** proved above. Now, we prove the *iff* condition for the perfect dependent part in the following two opposite directions.

"$\Longrightarrow$": If $\alpha_i = 1$, then $\Pr\{X_{i+1} = 0, X_i = 0\} \equiv \Pr\{X_i = 0\}\Pr\{X_{i+1} = 0 \mid X_i = 0\} = \Pr\{X_i = 0\}\, w^{(i)} \mid_{\alpha_i=1} = \Pr\{X_i = 0\} \left(\dfrac{1 - p_{i+1}}{1 - p_i}\right) = (1 - p_i)\left(\dfrac{1 - p_{i+1}}{1 - p_i}\right) = 1 - p_{i+1} = \Pr\{X_{i+1} = 0\}$. This says $\{X_i = 0\}$ and $\{X_{i+1} = 0\}$ are "perfectly" dependent, and in fact $\{X_{i+1} = 0\}$ is proper sub-event of $\{X_i = 0\}$ because $p_i < p_{i+1}$ implies $\Pr\{X_{i+1} = 0\} = 1 - p_{i+1} < 1 - p_i = \Pr\{X_i = 0\}$.

"$\Longleftarrow$": If $\{X_i = 0\}$ and $\{X_{i+1} = 0\}$ are "perfectly" dependent, then $\{X_{i+1} = 0\}$ is proper sub-event of $\{X_i = 0\}$ because $p_i < p_{i+1}$, i.e., $\Pr\{X_{i+1} = 0\} = 1 - p_{i+1} < 1 - p_i = \Pr\{X_i =$

$0\}$. So, $\Pr\{X_{i+1} = 0, X_i = 0\} = \Pr\{X_{i+1} = 0\} = (1 - p_{i+1})$. But $\Pr\{X_{i+1} = 0, X_i = 0\} \equiv \Pr\{X_i = 0\}\Pr\{X_{i+1} = 0 \mid X_i = 0\} = (1 - p_{i+1}) \equiv (1 - p_i)\left(\frac{1 - p_{i+1}}{1 - p_i}\right) = \Pr\{X_i = 0\}\left(\frac{1 - p_{i+1}}{1 - p_i}\right)$. This implies that $\Pr\{X_{i+1} = 0 \mid X_i = 0\} = \frac{1 - p_{i+1}}{1 - p_i} = w^{(i)} \mid_{\alpha_i = 1}$ for $p_i < p_{i+1}$ according to Eq. (U.24). Thus, we obtain $\alpha_i = 1$, which proves the second part of Eq. (5.17). This completes the proof of Eq. (5.17) for the case of $x_{i+1} = 0$ and $x_i = 0$. Eq. (5.18) can be proved in a similar way used to prove Eq. (5.17).

Claim 3: Eqs. (5.19) – (5.22) follow by plugging Eqs. (5.13) – (5.16) into Eq. (U.19). Hence the proof follows. ■

APPENDIX V

PROOF OF THEOREM 5.4.1

Proof. Claim 1: It follows directly from Claim 2 of Corollary 5.3.1 by letting $p_i = p'_i = p$ and $\alpha_i = \alpha'_i = \alpha \ \forall i \in \{1, 2, \cdots\}$.

Claim 2: Since $\psi_d(P_k, \alpha, p, m)$ is a real-valued continuous function of p and differentiable for $0 < p < 1$, we can take a partial derivative of $\psi_d(P_k, \alpha, p, m)$ with respect to p and set it to zero. For different ranges of k, we have the following three cases.

CASE 1. $k = 1$: Noticing that $\psi_d(P_k, \alpha, p, m)\mid_{k=1}$ is a strictly increasing function of p for $\alpha \in [0, 1]$, and $\psi_d(P_k, \alpha, p, m)\mid_{(k=1,\, p=1)} = 1$, i.e., $\psi_d(P_k, \alpha, p, m)\mid_{k=1}$ attains the maximum value at the boundary point $p = 1$, we obtain $p^* = 1$, thus yielding the first part of Eq. (5.27).

CASE 2. $k = m$:

$$\begin{aligned} \left.\frac{\partial \psi_d(P_k, \alpha, p, m)}{\partial p}\right|_{k=m} &= (1-\alpha)\Big\{(1-2p)[1-(1-\alpha)p]^{2m-3} \\ &\quad -(1-\alpha)(2m-3)p(1-p)[1-(1-\alpha)p]^{2m-4}\Big\} = 0 \end{aligned} \tag{V.1}$$

Solving Eq. (V.1) for the meaningful solution (root) with respect to p under the constraint of $0 < p < 1$, we obtain:

$$p^* = \arg \max_{0<p<1} \psi_d(P_k, p, m) = \frac{m-(m-1)\alpha-\sqrt{[m-(m-1)\alpha]^2-(1-\alpha)(2m-1)}}{(1-\alpha)(2m-1)} \tag{V.2}$$

which is unique and gives the second part of the Eq. (5.27).

CASE 3. $2 \leq k \leq m-1$:

$$\frac{\partial \psi_d(P_k, \alpha, p, m)}{\partial p} = [(1-2p)(2-p+\alpha p)-(1-\alpha)(1-p)p][1-(1-\alpha)p]^{2k-3}$$
$$-(1-\alpha)(2k-3)(1-p)p[2-(1-\alpha)p][1-(1-\alpha)p]^{2k-4} = 0 \quad \text{(V.3)}$$

Reducing Eq. (V.3), we get Eq. (5.28).

Claim 3: Since $\psi_d(P_k, \alpha, p, m)$ is a real-valued continuous function of α and differentiable for $0 \leq \alpha \leq 1$, we can take a partial derivative of $\psi_d(P_k, \alpha, p, m)$ with respect to α and set it to zero. For different ranges of k, we have the following two cases.

CASE 1. $2 \leq k \leq m-1$ and $k \geq \left\lceil \frac{1}{2} + \frac{1}{p(2-p)} \right\rceil$:

$$\frac{\partial \psi_d(P_k, \alpha, p, m)}{\partial \alpha} = [p(1-\alpha)-(2-p+\alpha p)][1-(1-\alpha)p]^{2k-3}$$
$$+(2k-3)(1-\alpha)p[2-(1-\alpha)p][1-(1-\alpha)p]^{2k-4} = 0 \quad \text{(V.4)}$$

Solving Eq. (V.4) for the root with respect to α which is unique and noticing $\alpha \geq 0$, we obtain:

$\alpha^* = \frac{p-1}{p} + \frac{1}{p}\sqrt{1-\frac{2}{2k-1}}$; but $\alpha^* \geq 0$ and k must be an integer, so we have:

$$k \geq \left\lceil \frac{1}{2} + \frac{1}{p(2-p)} \right\rceil,$$

which gives the first part of Eq. (5.29).

CASE 2. $k = m$ and $k \geq \left\lceil 1 + \frac{1}{2p} \right\rceil$:

$$\left.\frac{\partial \psi_d(P_k, \alpha, p, m)}{\partial \alpha}\right|_{k=m} = (1-p)p\Big\{(2m-3)(1-\alpha)p - [1-(1-\alpha)p]\Big\} = 0 \quad \text{(V.5)}$$

Solving Eq. (V.5) for the root with respect to α which is unique and noticing $0 \leq \alpha \leq 1$, we obtain:

$\alpha^* = 1 - \frac{1}{2(m-1)p}$; but $\alpha^* \geq 0$ must hold and m must be an integer, thus we have:

$$m \geq \left\lceil 1 + \frac{1}{2p} \right\rceil,$$

which is the second part of Eq. (5.29) as $m = k$.

<u>Claim 4</u>: Letting $\psi_d(P_k, \alpha, p, m) \mid_{\alpha=0} = \psi_d(P_k, \alpha, p, m) \mid_{\alpha=\alpha_0}$, we get

$$p(2-p)(1-p)^{2k-2} = (1-\alpha_0)(1-p)p[2-(1-\alpha_0)p][1-(1-\alpha_0)p]^{2k-3} \tag{V.6}$$

Solving Eq. (V.6) for the root with respect to k which is unique, we obtain:

$$\widetilde{k} = \frac{\log\sqrt{\dfrac{2-p}{(1-\alpha_0)[2-(1-\alpha_0)p]}}}{\log\dfrac{1-(1-\alpha_0)p}{1-p}} + 1.5, \tag{V.7}$$

Taking the integer part of $\widetilde{k}$, we obtain Eq. (5.32).

On the other hand, because $m < \infty$, the multicast-tree height m must be large enough to ensure the existence of $\widetilde{k}$. To derive the lower bound of m, let $k = m$ and also $\psi_d(P_m, \alpha, p, m) \mid_{\alpha=0} = \psi_d(P_m, \alpha, p, m) \mid_{\alpha=\alpha_0}$, we get

$$p(1-p)^{2m-2} = (1-\alpha_0)(1-p)p[1-(1-\alpha_0)p]^{2m-3} \tag{V.8}$$

Solving Eq. (V.8) for the root with respect to m, we obtain:

$$\widetilde{m} = \left\lfloor \frac{\log\sqrt{\dfrac{1}{(1-\alpha_0)}}}{\log\dfrac{1-(1-\alpha_0)p}{1-p}} + 1.5 \right\rfloor, \quad \Longrightarrow \quad m \geq \widetilde{m} + 1 = \left\lfloor \frac{\log\sqrt{\dfrac{1}{(1-\alpha_0)}}}{\log\dfrac{1-(1-\alpha_0)p}{1-p}} + 2.5 \right\rfloor, \tag{V.9}$$

where the inequality of Eq.(V.9) is because $2 \leq \widetilde{k} \leq m-1$, and we need $m \geq \widetilde{m}+1 \geq \widetilde{k}+1$. Thus Eq. (5.30) follows.

Now, we prove Eq. (5.31) in the following two cases:

<u>*CASE 1:*</u> $k \leq \widetilde{k}$. By Eq. (V.7), we have

$$k \leq \widetilde{k} = \frac{\log\sqrt{\dfrac{2-p}{(1-\alpha_0)[2-(1-\alpha_0)p]}}}{\log\dfrac{1-(1-\alpha_0)p}{1-p}} + 1.5, \tag{V.10}$$

Reducing Eq. (V.10), we obtain:

$$(1-\alpha_0)(1-p)p[2-(1-\alpha_0)p][1-(1-\alpha_0)p]^{2k-3} \leq p(1-p)(2-p)(1-p)^{2k-3}, \tag{V.11}$$

where, according to Eq. (5.26), the left-hand side of Eq. (V.11) is $\psi_d(P_k, \alpha, p, m) \mid_{\alpha=\alpha_0}$ and the right-hand side of Eq. (V.11) is $\psi_d(P_k, \alpha, p, m) \mid_{\alpha=0}$. Thus, the first part of Eq. (5.31) follows.

<u>*CASE 2:*</u> $k > \widetilde{k}$. Also by Eq. (V.7), we have

$$k > \widetilde{k} = \frac{\log \sqrt{\dfrac{2-p}{(1-\alpha_0)[2-(1-\alpha_0)p]}}}{\log \dfrac{1-(1-\alpha_0)p}{1-p}} + 1.5, \tag{V.12}$$

Reducing Eq. (V.12), we obtain

$$(1-\alpha_0)(1-p)p[2-(1-\alpha_0)p][1-(1-\alpha_0)p]^{2k-3} > p(1-p)(2-p)(1-p)^{2k-3}, \tag{V.13}$$

where, according to Eq. (5.26), the left-hand side of Eq. (V.13) is $\psi_d(P_k, \alpha, p, m) \mid_{\alpha=\alpha_0}$ and the right-hand side of Eq. (V.13) is $\psi_d(P_k, \alpha, p, m) \mid_{\alpha=0}$. Thus, the second part of Eq. (5.31) follows.

<u>**Claim 5**</u>**:** Eqs. (5.34) and (5.33) follow by plugging Eq. (5.26), and Theorem 3.4.1's Eq. (3.1) and Theorem 3.4.2's Eq. (3.6), respectively, into Eq. (V.14) that follows below:

$$\overline{\tau}(m) = E[\tau(m)] = \sum_{j=1}^{m} \tau_u(j, \Delta)\psi_d(P_j, \alpha, p, m) \tag{V.14}$$

<u>**Claim 6**</u>**:** Eqs. (5.36) and (5.35) follow by plugging Eq. (5.26), and Theorem 3.4.1's Eq. (3.1) and Theorem 3.4.2's Eq. (3.6), respectively, into Eq. (V.15) that follows below:

$$\begin{aligned}\sigma^2(m) = Var[\tau(m)] &= \sum_{j=1}^{m}[\tau_u(j, \Delta)]^2\psi_d(P_j, \alpha, p, m) \\ &\quad - \left(\sum_{j=1}^{m} \tau_u(j, \Delta)\psi_d(P_j, \alpha, p, m)\right)^2 \end{aligned} \tag{V.15}$$

This completes the proof. ■

APPENDIX W

PROOF OF THEOREM 5.5.1

Proof. Claim 1: Since the link-marking probability vector $\vec{p} = (p_1, p'_1, p_2, p'_2, p_3, p'_3, \cdots)$ defined in Definition 5.2.1 and $\vec{\alpha}$ satisfy $0 < p_i = p'_i = p < 1$ and $0 \leq \alpha_i = \alpha'_i = \alpha \leq 1$, $\forall i$, respectively, $\{X_i\}$ becomes a homogeneous Markov chain, and its one-step state transition probabilities are fixed and independent of link numbers. The matrix P of one-step transition probabilities, which is defined by Eqs. (5.19) through (5.22) for $\alpha_i = \alpha$ and $p_i = p$, $\forall i \in \{1, 2, \cdots\}$, is given by:

$$P \triangleq \{p_{jk}\} \triangleq \Big\{\Pr\{X_{i+1} = k \mid X_i = j\}\Big\} = \begin{bmatrix} 1-(1-\alpha)p & (1-\alpha)p \\ \\ (1-\alpha)(1-p) & \alpha(1-p)+p \end{bmatrix} \tag{W.1}$$

where $j, k \in \{0, 1\}$ and $\forall i \in \{1, 2, \cdots\}$. Now, we prove Eq. (Y.11) for cases of $n \in \{1, 2, \cdots\}$[1] by mathematical induction.

[1] It is easy to prove that Eq. (Y.11) also holds for the trivial case of $n = 0$. Based on the definition of 0-step link-marking state transition probability, we have the following transition-probability expression:

$$p_{jk}^{(0)} = \Pr\{X_r = k \mid X_r = j\} = \begin{cases} 1, & \text{if } j = k; \\ 0, & \text{if } j \neq k; \end{cases} \tag{W.2}$$

where $j, k \in \{0, 1\}$ and $\forall r \in \{1, 2, \cdots\}$. So, Eq. (W.2) yields a 2×2 unit matrix I. On the other hand, according to Eq. (Y.11), we have $P^{(0)} = I$, which is also a 2×2 unit matrix. Thus, Eq. (Y.11) also holds for $n = 0$.

Base Case: $n = 1$. By Eq. (W.1), we have

$$P = \begin{bmatrix} 1-(1-\alpha)p & (1-\alpha)p \\ \\ (1-\alpha)(1-p) & \alpha(1-p)+p \end{bmatrix} = P^{(n)}\big|_{n=1} \tag{W.3}$$

where $P^{(n)}$ is defined in Eq. (Y.11). Thus, Eq. (Y.11) holds for $n = 1$.

Inductive Hypothesis: Suppose Eq. (Y.11) holds for $n = q-1$, i.e.,

$$P^{(q-1)} \triangleq \left\{p_{jk}^{(q-1)}\right\} = \begin{bmatrix} 1-(1-\alpha^{(q-1)})p & (1-\alpha^{(q-1)})p \\ \\ (1-\alpha^{(q-1)})(1-p) & \alpha^{(q-1)}(1-p)+p \end{bmatrix} \tag{W.4}$$

and we need to prove that it also holds for $n = q$ as follows:

$$\begin{aligned} P^{(q)} \triangleq \left\{p_{jk}^{(q)}\right\} = P^{(q-1)}P \quad &= \begin{bmatrix} 1-(1-\alpha^{(q-1)})p & (1-\alpha^{(q-1)})p \\ \\ (1-\alpha^{(q-1)})(1-p) & \alpha^{(q-1)}(1-p)+p \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} 1-(1-\alpha)p & (1-\alpha)p \\ \\ (1-\alpha)(1-p) & \alpha(1-p)+p \end{bmatrix} \end{aligned} \tag{W.5}$$

where P is defined by Eq. (W.1). With a little algebra, Eq. (W.5) reduces to

$$P^{(q)} \triangleq \left\{p_{jk}^{(q)}\right\} = \begin{bmatrix} 1-(1-\alpha^{q})p & (1-\alpha^{q})p \\ \\ (1-\alpha^{q})(1-p) & \alpha^{q}(1-p)+p \end{bmatrix} \tag{W.6}$$

Thus, Eq. (Y.11) holds for $n = q$. So, in general Eq. (Y.11) holds for $n \in \{0, 1, 2, \cdots\}$.

Claim 2: Eq. (5.39) follows directly from Eq. (Y.11). To prove state j ($\in \{0,1\}$) is ergodic, we need to prove that j is positive recurrent and aperiodic. Clearly, j is aperiodic (period $d = 1$), but we need to prove j is positive recurrent which is true *iff* the following two conditions hold:

$$\underline{\text{Condition 1}}: \ \limsup_{n\to\infty} \ p_{jj}^{(n)} > 0, \quad \text{and} \quad \underline{\text{Condition 2}}: \ \lim_{n\to\infty} \sum_{r=1}^{n} p_{jj}^{(r)} = \infty, \tag{W.7}$$

But because j is aperiodic, we have $\limsup_{n\to\infty} p_{jj}^{(n)} = \lim_{n\to\infty} p_{jj}^{(n)}$. From Eq. (5.39), due to $0 < p < 1$ it follows that

$$\limsup_{n\to\infty} p_{jj}^{(n)} = \lim_{n\to\infty} p_{jj}^{(n)} > 0 \tag{W.8}$$

for $\alpha \in [0,1]$, which proves the Condition 1 in Eq. (W.7). On the other hand, the Condition 2 in Eq. (W.7) also holds because from Eq. (Y.11), we have

$$\lim_{n\to\infty} \sum_{r=1}^{n} p_{00}^{(r)} = \sum_{n=1}^{\infty}(1-p) + \sum_{n=1}^{\infty} p\alpha^n = \infty, \tag{W.9}$$

and

$$\lim_{n\to\infty} \sum_{r=1}^{n} p_{11}^{(r)} = \sum_{n=1}^{\infty}(1-p)\alpha^n + \sum_{n=1}^{\infty} p = \infty, \tag{W.10}$$

where $0 < p < 1$ and $\alpha \in [0,1]$. Thus, the two states of the Markov chain $\{X_i\}$ are ergodic.

Claim 3: If $\alpha \in [0,1)$, i.e., $\alpha \neq 1$, implying $\{X_i\}$ is not perfectly dependent, then $\{X_i\}$ is irreducible, and further the Markov chain $\{X_i\}$ is ergodic because it is positive recurrent and aperiodic as proved in **Claim 2** above. Then the ergodic Markov chain $\{X_i\}$ has the unique limiting (equilibrium) state probabilities which are determined by Eq. (Y.11) as follows:

$$\pi_0 = \lim_{n\to\infty} p_{j0}^{(n)} = 1 - p = \Pr\{X_i = 0\} \quad \text{and} \quad \pi_1 = \lim_{n\to\infty} p_{j1}^{(n)} = p = \Pr\{X_i = 1\} \tag{W.11}$$

where $\alpha \in [0,1)$ and $\forall j \in \{0,1\}$. Hence Eq. (5.40) follows.

Claim 4: If $\alpha = 1$, i.e., $\{X_i\}$ is perfectly dependent, then $\{X_i\}$ has two isolated states (see Figure 5.3 where the transition probabilities between the two states become 0 when $\alpha = 1$). So, $\{X_i\}$ is not irreducible anymore, and thus is not ergodic. Furthermore, Eq. (Y.11) shows that the n-step probability matrix $P^{(n)}$ reduces to a 2×2 unit matrix I when $\alpha = 1$. Thus, the equilibrium state probabilities of $\{X_i\}$ are determined by

$$\begin{aligned} \vec{\pi} &\triangleq \begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} = \lim_{n\to\infty} \vec{p}(n) = \lim_{n\to\infty} \begin{bmatrix} p_0(1) & p_1(1) \end{bmatrix} P^{(n-1)} = \lim_{n\to\infty} \begin{bmatrix} p_0(1) & p_1(1) \end{bmatrix} I \\ &= \begin{bmatrix} p_0(1) & p_1(1) \end{bmatrix} \end{aligned} \tag{W.12}$$

where $\vec{p}(n) \triangleq \left[\begin{array}{cc} p_0(n) & p_1(n) \end{array} \right] = \left[\begin{array}{cc} p_0(1) & p_1(1) \end{array} \right] P^{(n-1)}$ denotes the vector of state probabilities at link n (note that $n \geq 1$ because the Markov-chain's index set consists of the link sequence numbers n instead of time variables and thus the initial value of n is 1; also note that by Eq. (Y.11), $P^{(0)} = I$ which is a 2×2 unit matrix), and $\left[\begin{array}{cc} p_0(1) & p_1(1) \end{array} \right]$ are the initial state probabilities of the two states, which are generally arbitrary and thus not unique. However, since the initial state ($n = 1$) probabilities of each state are equal to the marginal link-marking probabilities in the cases addressed in this chapter, so

$$\vec{\pi} \triangleq \left[\begin{array}{cc} \pi_0 & \pi_1 \end{array} \right] = \left[\begin{array}{cc} p_0(1) & p_1(1) \end{array} \right] = \left[\begin{array}{cc} \Pr\{X_i = 0\} & \Pr\{X_i = 1\} \end{array} \right] = \left[\begin{array}{cc} 1-p & p \end{array} \right]$$

still holds. This completes the proof. ∎

APPENDIX X

PROOF OF THEOREM 6.2.1

Proof. Since the primal-optimization problem's objective functions, given by Eq (6.4) and Eq (6.4) for $\mathbf{P}$ and $\mathbf{P}^*$, respectively, are exactly the same, we only need to prove that the optimal solutions for $\mathbf{P}$ and $\mathbf{P}^*$ are the same. Letting $\mathcal{F}$ and $\mathcal{F}^*$ be the feasible solution sets of the primal-optimization problem $\mathbf{P}$ and $\mathbf{P}^*$, respectively, we have

$$\mathcal{F} = \left\{ (r_1, r_2, \cdots, r_S) \mid \sum_{s \in \mathcal{S}(\ell)} r_s - \mu_\ell \leq 0, \quad \forall \ell \in \mathcal{L} = \{1, 2, \cdots, L\} \right\} \tag{X.1}$$

and

$$\mathcal{F}^* = \left\{ (r_1, r_2, \cdots, r_S) \mid \sum_{s \in \mathcal{S}(\ell)} r_s - \mu_\ell \leq 0, \quad \forall \ell \in \mathcal{L}^* = \{1, 2, \cdots, L^*\} \right\}. \tag{X.2}$$

Using the definition of $\mathcal{F}^*$ given by Eq. (6.3) and combining Eq. (X.1) and Eq. (X.2), we obtain

$$\mathcal{L}^* \subseteq \mathcal{L} \qquad \Longrightarrow \qquad \mathcal{F}^* \subseteq \mathcal{F}. \tag{X.3}$$

On the other hand, according to the definitions of $\mathbf{P}$ and $\mathbf{P}^*$ and their constraints given by Eqs (6.2) and (6.5), respectively, the primal-optimal solution of $\mathbf{P}$ and $\mathbf{P}^*$ must be determined by the *most congested* paths which are defined by Eq. (6.3) and condition **C4** in Definition 6.2.1. Therefore, the primal-optimization solutions for both $\mathbf{P}$ and $\mathbf{P}^*$ must lie in $\mathcal{F}^*$. This implies that the primal-optimization solutions for both $\mathbf{P}$ and $\mathbf{P}^*$ are equal by Eq. (X.3), completing the proof. ■

APPENDIX Y

PROOF OF THEOREM 6.2.2

Proof. Claim 1: Applying the Lagrange Duality Theory [80, 81, 106] to the primal-optimization problem $\mathbf{P}^*$ specified by Theory 6.2.1, we define the Lagrangian function $L(\vec{r}, \vec{\lambda}_*)$: $\Re^{S+L^*} \longmapsto \Re$, as follows:

$$L(\vec{r}, \vec{\lambda}_*) \triangleq \sum_{s \in \mathcal{S}} U_s(r_s) - \sum_{\ell \in \mathcal{L}^*} \lambda_\ell \left(\sum_{s \in \mathcal{S}(\ell)} r_s - \mu_\ell \right) \tag{Y.1}$$

where $\vec{\lambda}_* = (\lambda_1, \cdots, \lambda_{L^*})$ is the Lagrange-multiplier vector and $\left(\sum_{s \in \mathcal{S}(\ell)} r_s - \mu_\ell\right) \leq 0$, $\forall \ell \in \mathcal{L}^*$, is the constraint vector specified in Definition 6.2.2. Noting the fact that

$$\sum_{\ell \in \mathcal{L}^*} \sum_{s \in \mathcal{S}(\ell)} \lambda_\ell \, r_s = \sum_{s \in \mathcal{S}} \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell \, r_s \tag{Y.2}$$

we can reduce Eq. (Y.1), which leads to

$$L(\vec{r}, \vec{\lambda}_*) = \sum_{s \in \mathcal{S}} \left(U_s(r_s) - r_s \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell \right) + \sum_{\ell \in \mathcal{L}^*} \lambda_\ell \mu_\ell \tag{Y.3}$$

The objective function of the dual optimization problem can thus be obtained [80,81] through the Lagrangian function as follows:

$$
\begin{aligned}
D^*(\vec{\lambda}_*) &\triangleq \max_{r_s \in I_s, s \in \mathcal{S}} L(\vec{r}, \vec{\lambda}_*) && \text{(Y.4)} \\
&= \max_{r_s \in I_s, s \in \mathcal{S}} \left\{ \sum_{s \in \mathcal{S}} \left(U_s(r_s) - r_s \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell \right) + \sum_{\ell \in \mathcal{L}^*} \lambda_\ell \mu_\ell \right\} && \text{(Y.5)} \\
&= \sum_{s \in \mathcal{S}} \max_{r_s \in I_s, s \in \mathcal{S}} \left\{ U_s(r_s) - r_s \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell \right\} + \sum_{\ell \in \mathcal{L}^*} \lambda_\ell \mu_\ell && \text{(Y.6)} \\
&= \sum_{s \in \mathcal{S}} B_s^*(\lambda_{k^*}^s) + \sum_{\ell \in \mathcal{L}^*} \lambda_\ell \mu_\ell && \text{(Y.7)}
\end{aligned}
$$

where

$$
B_s^*(\lambda_{k^*}^s) \triangleq \max_{r_s \in I_s, s \in \mathcal{S}} \{U_s(r_s) - r_s \lambda_{k^*}^s\} \quad \text{and} \quad \lambda_{k^*}^s \triangleq \sum_{\ell \in \mathcal{L}_{k^*}(s)} \lambda_\ell. \qquad \text{(Y.8)}
$$

Then, the dual optimization problem for the multicast flow control is

$$
\mathbf{D}^*: \quad \min_{\lambda_\ell \geq 0,\, \ell \in \mathcal{L}^*} D^*(\vec{\lambda}_*) \qquad \text{(Y.9)}
$$

which yields the desired result for Claim 1.

Claim 2: Since $U_s(r_s)$ is chosen to be strictly concave, ensuring that $U_s(r_s)$ is continuous, and the feasible solution set is compact, $\mathbf{P}^*$'s primal-optimal solution exists and is unique. Therefore, according to the duality theory, the $\mathbf{D}^*$'s dual-optimal solution also exists, is unique, and equals the $\mathbf{P}^*$'s primal-optimal solution.

Claim 3: Since $\mathbf{D}^*(\vec{\lambda}_*)$ is transformed to a non-constrained optimization problem by using Lagrange multiplier, we apply the Gradient Projection method [81, 107] to solve the standard minimization problem $\mathbf{D}^*(\vec{\lambda}_*)$, where the Lagarangian multipliers at all links are adjusted in direction opposite to the gradient $\nabla D^*(\vec{\lambda}_*)$, that is, at link $\ell \in \mathcal{L}^*$, the Lagarangian multiplier λ_ℓ at next time instance $(t+1)$ is updated iteratively as follows:

$$
\lambda_\ell(t+1) = \left[\lambda_\ell(t) - \gamma \frac{\partial D^*}{\partial \lambda_\ell} \left(\vec{\lambda}_*(t) \right) \right]^+ . \qquad \text{(Y.10)}
$$

The gradient vector $\nabla D^*(\vec{\lambda}_*)$ of the dual objective function is determined by

$$\nabla D^*(\vec{\lambda}) \triangleq \begin{pmatrix} \frac{\partial D^*(\vec{\lambda}(t))}{\partial \lambda_1} \\ \\ \frac{\partial D^*(\vec{\lambda}(t))}{\partial \lambda_2} \\ \\ \vdots \\ \\ \frac{\partial D^*(\vec{\lambda}(t))}{\partial \lambda_{L^*}} \end{pmatrix} \tag{Y.11}$$

where each term in the column vector given by Eq. (Y.11) can be derived as follows. Using Eq. (Y.6), for $\forall s \in \mathcal{S}$, we obtain the gradient of the dual objective function at time t at link $\ell \in \mathcal{L}_{k^*}(s)$:

$$\begin{aligned} \frac{\partial D^*\left(\vec{\lambda}_*(t)\right)}{\partial \lambda_\ell} &= \frac{\partial}{\partial \lambda_\ell}\left[\sum_{s\in\mathcal{S}} \max_{r_s\in I_s, s\in\mathcal{S}} \left\{ U_s\left(r_s\left(\vec{\lambda}_*(t)\right)\right) - r_s\left(\vec{\lambda}_*(t)\right) \sum_{\ell\in\mathcal{L}_{k^*}(s)} \lambda_\ell \right\}\right. \\ &\qquad \left. + \sum_{\ell\in\mathcal{L}^*} \lambda_\ell \mu_\ell \right] \qquad (Y.12) \\ &= \frac{\partial}{\partial \lambda_\ell} \sum_{\ell\in\mathcal{L}^*} \lambda_\ell\, \mu_\ell - \sum_{s\in\mathcal{S}} r_s\left(\vec{\lambda}_*(t)\right) \frac{\partial}{\partial \lambda_\ell} \sum_{\ell\in\mathcal{L}_{k^*}(s)} \lambda_\ell \\ &= \mu_\ell - \sum_{s\in\mathcal{S}(\ell),\, \ell\in\mathcal{L}_{k^*}(s)} r_s\left(\vec{\lambda}_*(t)\right), \quad \forall \ell \in \mathcal{L}^* = \{1, 2, \cdots, L^*\} \qquad (Y.13) \end{aligned}$$

where Eq. (Y.13) holds because for $s' \notin \mathcal{S}(\ell)$, $\mathcal{L}_{k^*}(s')$ is not a subset of $\mathcal{L}_{k^*}(s) \Longrightarrow \forall$ $\ell' \in \mathcal{L}_{k^*}(s')$, we have $\ell' \notin \mathcal{L}_{k^*}(s) \Longrightarrow \ell' \neq \ell$, thus $\sum_{s\notin\mathcal{S}(\ell)} \frac{\partial \lambda_{\ell'}}{\partial \lambda_\ell} = 0$. Plugging Eq. (Y.13) into Eq. (Y.10) to calculate the new target value of Lagrange Multiplier for the next iterative step at time $(t+1)$, we have

$$\lambda_\ell(t+1) = \left[\lambda_\ell(t) \;+\; \gamma\left(\sum_{s\in\mathcal{S}(\ell),\, \ell\in\mathcal{L}_{k^*}(s)} r_s\left(\vec{\lambda}_*(t)\right) - \mu_\ell\right)\right]^+, \quad \forall \ell \in \mathcal{L}_{k^*}(s),$$

which proves Eq. (6.11) of Claim 3.

Claim 4: The sumation term $\sum_{s\in\mathcal{S}} B_s^*(\lambda_{k^*}^s)$ of Eq. (6.8) shows that the dual-optimization problem $\mathbf{D}^*$ is decomposed into S separable subproblems. Once the minimizer vector of

Lagarangian multiplier $\vec{\lambda}^o_*$ is obtained by solving Eq. (6.7), the primal-optimal multicast flow-control rates $\vec{r}_o$ can be obtained by each individual multicast-traffic source s by solving Eq. (6.9) and Eq. (6.10), which is a simple maximization and can be implemented locally based only on local flow-control information. This proves that $\mathbf{D}^*$ decomposes the primal-optimal multicast flow-control problem $\mathbf{P}^*$ into S independent subproblems in terms of the objective aggregate-utility function.

On the other hand, Eq. (6.11) shows that the bandwidth constraints functions are also decomposed because each link $\ell \in \mathcal{L}^*$ only needs to know the local bandwidth contraint μ_ℓ, without needing to know $\mu_{\ell'}$ ($\ell' \neq \ell$) of other link-bandwidth constraints, to iteratively search for the local optimal Lagarangian multiplier λ_ℓ, as shown in Eq. (6.11). Thus, $\mathbf{D}^*$ also decomposes the primal-optimal multicast flow-control problem in terms of the aggregate constraints. This completes the proof of Theorem 6.2.2. ∎

APPENDIX Z

PROOF OF THEOREM 6.4.1

Proof. Using the definition of F_η given in Eq. (6.19) and observing that ECN-bit markings are all independent, we obtain

$$\begin{aligned}
E\left[F_\eta\right] &= \sum_{y_1=0}^{N}\sum_{y_2=0}^{N}\cdots\sum_{y_n=0}^{N}\left\{1-\frac{1}{N}\max\{Y_1=y_1,Y_2=y_2,\cdots,Y_n=y_n\}\right.\\
&\quad \left.\cdot\exp\left(-\left[N-\max\{Y_1=y_1,Y_2=y_2,\cdots,Y_n=y_n\}\right]\right)\right\}\\
&\quad \cdot\prod_{j=1}^{n}\binom{N}{y_j}p_j^{y_j}\left(1-p_j\right)^{N-y_j}\\
&= \sum_{i=0}^{N-1}\left\{\left\{1-\frac{i}{N}e^{-\frac{1}{n}[N-i]}\right\}\right.\\
&\quad \left.\cdot\sum_{\{(y_1,y_2,\cdots,y_n)\,|\,\max_{1\le j\le n}\{y_j\}=i\}}\prod_{j=1}^{n}\binom{N}{y_j}p_j^{y_j}\left(1-p_j\right)^{N-y_j}\right\}
\end{aligned} \tag{Z.1}$$

which yields the first line of Eq. (6.22). All the other equations for $Var\left[F_\eta\right]$, $E\left[F_\alpha\right]$, $Var\left[F_\alpha\right]$, $E\left[F_\mu\right]$, and $Var\left[F_\mu\right]$ of this theorem can be proved similarly, thus completing the proof. ■

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Van Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM*, 1988.

[2] Xi Zhang, Kang G. Shin, Debajan Saha, and Dilip D. Kandlur, "Scalable flow control for multicast ABR services in ATM networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, , February 2002.

[3] Xi Zhang and Kang G. Shin, "Delay analysis of feedback-synchronization signaling for multicast flow control," *IEEE/ACM Transactions on Networking*, Accepted for publication subject to minor revision, 2001.

[4] Xi Zhang and Kang G. Shin, "Markov-chain modeling for multicast signaling delay analysis," *IEEE/ACM Transactions on Networking*, Accepted for publication subject to minor revision, 2001.

[5] Xi Zhang and Kang G. Shin, "Statistical analysis of feedback synchronization signaling delay for multicast flow control," in *Proc. of IEEE INFOCOM*, pp. 1152–1161, April 2001.

[6] Xi Zhang and Kang G. Shin and Debenjan Saha and Dilip D. Kandlur, "Scalable flow control for multicast ABR services," in *Proc. of IEEE INFOCOM*, pp. 837–846, March 1999.

[7] Xi Zhang and Kang G. Shin and Qin Zheng, "Integrated rate and credit feedback control for ABR services in ATM networks," in *Proc. of IEEE INFOCOM*, pp. 1297–1305, April 1997.

[8] Xi Zhang and Kang G. Shin, "Second-order rate-control based transport protocols," in *Proc. of IEEE International Conference on Network Protocols (ICNP)*, pp. 342–350, November 2001.

[9] Xi Zhang and Kang G. Shin, "Optimization-based multicast flow control using virtual M-Ary feedback," *submitted to ACM SIGCOMM'02 for publication*, February 2002.

[10] Xi Zhang and Kang G. Shin, "Virtual M-Ary feedback-based optimization multicast flow control using binary feedback," *Technical Report, Real-Time Computing Laboratory, EECS Dept., The University of Michigan, Ann Arbor*, November 2001.

[11] Xi Zhang and Kang G. Shin, "Performance analysis of feedback synchronization for multicast ABR flow control," in *Proc. of IEEE GLOBECOM*, pp. 1269–1274, December 1999.

[12] Xi Zhang and Kang G. Shin, "A scalable flow-control algorithm for point-to-multipoint communications in high-speed integrated networks," *Technical Report, CSE-TR-365-98, EECS Dept., The University of Michigan, Ann Arbor*, June 1998.

[13] Jon Crowcroft and Ken Paliwoda, "A multicast transport protocol," in *Proc. of ACM SIGCOMM*, pp. 247–256, August 1988.

[14] Xi Zhang and Kang G. Shin, "Second-order rate-based flow control with decoupled error control for high-throughput transport protocols," *Technical Report, CSE-TR-406-99, EECS Dept., The University of Michigan, Ann Arbor*, June 1999.

[15] W. Ren, K.-Y. Siu, and H. Suzuki, "On the performance of congestion control algorithms for multicast ABR service in ATM," in *Proc. of IEEE ATM WORKSHOP*, August 1996.

[16] Larry Roberts, *Rate Based Algorithm for Point to Multipoint ABR Service*, ATM Forum contribution 94-0772, September 1994.

[17] Larry Roberts, *Point-to-Multipoint ABR Operation*, ATM Forum contribution 95-0834, August 1995.

[18] K.-Y. Siu and H.-Y. Tzeng, "On max-min fair congestion control for multicast ABR services in ATM," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 545–556, April 1997.

[19] H. Saito, K. Kawashima, H. Kitazume, A. Koike, M. Ishizuka, and A. Abe, "Performance issues in public ABR service," *IEEE Communications magazine*, vol. 11, pp. 40–48, November 1996.

[20] Y.-Z. Cho, S.-M. Lee, and M.-Y. Lee, "An efficient rate-based algorithm for point-to-multipoint ABR service," in *Proc. of IEEE GLOBECOM*, pp. 790–795, November 1997.

[21] S. Fahmy, R. Jain, R. Goyal, B. Vandalor, and S. Kalyanaraman, "Feedbackback consolidation algorithms for ABR point-to-mulipoint connections in ATM networks," in *Proc. of IEEE INFOCOM*, pp. 1004–1013, April 1998.

[22] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, pp. 1–14, 1989.

[23] S. Sathaye, *ATM Forum traffic management specifications Version 4.0*, ATM Forum contribution 95-0013R7.1, August 1995.

[24] N. Yin and M. G. Hluchyj, "On closed-loop rate control for ATM cell relay networks," in *Proc. of IEEE INFOCOM*, pp. 99–109, June 1994.

[25] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Analysis of rate-based congestion control for ATM networks," *ACM SIGCOMM Computer Communication Review*, vol. 25, pp. 60–72, April 1995.

[26] F. Bonomi, D. Mitra, and J. Seery, "Adaptive algorithms for feedback-based flow control in high-speed, wide-area ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1267–1283, September 1995.

[27] S. Keshav, *An Engineering Approach to Computer Networking — ATM Networks, the Internet, and the Telephone Network*, Addison-Wesley Publishing Company, 1997.

[28] J. Bolot and A. Shankar, "Dynamical behavior of rate-based flow control mechanism," *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 4, pp. 35–49, April 1990.

[29] M. Ritter, "Network buffer requirements of the rate-based control mechanism for ABR services," in *Proc. of IEEE INFOCOM*, pp. 1190–1197, March 1996.

[30] David D. Clark and Mark Lambert and Lixia Zhang, "NETBLT: A high throughput transport protocol," in *ACM SIGCOMM*, pp. 353–359, 1987.

[31] A. Heybey, "The network simulator," *Laboratory for Computer Science, Massachusetts Institute of Technology*, September 1990.

[32] Xi Zhang and Kang G. Shin, "Second-order rate-based flow control with decoupled error control for high-throughput transport protocols," *Full paper version:* URL http://www.eecs.umich.edu/~xizhang/papers/rw.pdf, July 2001.

[33] Xi Zhang and Kang G. Shin and Debanjan Saha and Dilip D. Kandlur, "Scalable flow control for multicast ABR services in ATM networks," *Technical Report, CSE-TR-353-97, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor*, November 1997.

[34] K.-Y. Siu and H.-Y. Tzeng, "Congestion control for multicast service in ATM networks," in *Proc. of IEEE GLOBECOM*, pp. 310–314, November 1995.

[35] D. Lapsley and S. Low, "Random early marking: for Internet congestion control," in *Proc. of IEEE GLOBECOM*, pp. 1747–1752, December 1999.

[36] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

[37] D. Lapsley and S. Low, "An optimization approach to ABR control," in *Proc. of IEEE International Conference on Communications*, June 1998.

[38] S. Athuraliya, S. Low, and D. Lapsley, "Random early marking," in *Proceedings of the First International Workshop on Quality of future Internet Services (QofIS'2000), Berlin, Germany*, September 2000.

[39] D. Lapsley and S. Low, "An IP implementation of optimization flow control," in *Proc. of IEEE GLOBECOM*, November 1998.

[40] S. Athuraliya, D. Lapsley, and S. Low, "An enhanced random early marking algorithm for Internet flow control," in *Proc. of IEEE INFOCOM*, pp. 1425–1434, March 2000.

[41] S. Floyd, "TCP and explicit congestion notification," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 5, pp. 10–23, October 1994.

[42] A. Heybey, *The Network Simulator*, Laboratory for Computer Science, Massachusetts Institute of Technology, October 1989.

[43] S. H. Low and D. Lapsley, "Optimization flow control — I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, December 1999.

[44] S. H. Low, "Optimization flow control with on-line measurement," in *Proc. of International Teletraffic Congress*, volume 16, pp. 237–249, June 1999.

[45] Xi Zhang and Kang G. Shin, "Feedback soft-synchronization for random-marking-based multicast flow control," *Technical Report, Real-Time Computing Labs., EECS Dept., The University of Michigan*, March 2000.

[46] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint communication: A survey of protocols, functions, and mechanisms," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 277–290, April 1997.

[47] B. J. Vickers, M. Lee, and T. Suda, "Feedback control mechanism for real-time multipoint video services," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 512–530, April 1997.

[48] B. Shacham and H. Yokota, "Admission control algorithms for multicast sessions with multiple streams," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 557–566, April 1997.

[49] S. Kasera, S. Bhattacharyya, M. Keaton, D. Kiwior, S. Zabele, J. Kurose, and D. Towsley, "Scalable fair reliable multicast using active services," *IEEE Network Magazine*, vol. 14, no. 1, pp. 48–57, January/February 2000.

[50] J. Gemmell, J. Gray, and E. Schoolerose, "Fcast multicast file distribution," *IEEE Network Magazine*, vol. 14, no. 1, pp. 58–69, January/February 2000.

[51] P. Moghe and I. Rubin, "Reserving for future clients in a multipoint application — why and how?," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 531–544, April 1997.

[52] L. Gong, "Enclaves: enabling secure collaboration over Internet," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 567–575, April 1997.

[53] C. Blum, P. Dubois, R. Molva, and O. Schaller, "A development and runtime platform for teleconferencing applications," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 576–577, April 1997.

[54] B. Shacham, "Preemptiion-based admission control in multimedia multiparty communication," in *Proc. of IEEE INFOCOM*, volume 2, pp. 827–834, April 1995.

[55] R. Aiello, E. Pagani, and G. P. Rossi, "Design of reliable multicast protocol," in *Proc. of IEEE INFOCOM*, pp. 73–81, April 1993.

[56] S. Paul, "Multicasting: Empowering the next-generation Internet," *IEEE Network Magazine*, vol. 14, no. 1, pp. 8–9, January/February 2000.

[57] K. Almeroth, "The evolution of multicast: from the MBone to interdomain multicast to Internet2 deployment," *IEEE Network Magazine*, vol. 14, no. 1, pp. 10–20, January/February 2000.

[58] B. Whetten and G. Taskale, "Reliable multicast transport protocol II," *IEEE Network Magazine*, vol. 14, no. 1, pp. 37–47, January/February 2000.

[59] C. Diot, B. N. Levine, B.Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP service and architecture," *IEEE Network Magazine*, vol. 14, no. 1, pp. 78–88, January/February 2000.

[60] D. Towsley, J. Kurose, and S. Pingali, "A comparison of sender-initiated and receiver-initiated reliable multicast transport protocols," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 398–406, April 1997.

[61] Y. Ofek and B. Yener, "Reliable concurrent multicast from bursty sources," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 434–444, April 1997.

[62] G. J. Armitage, "IP multicasting over ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 445–457, April 1997.

[63] E. Gauthier, J.-Y. L. Boudec, and P. Oechslin, "SMART a many-to-many multicast protocol for ATM," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 458–472, April 1997.

[64] F. M. Chiussi, Y. Xia, and V. P. Kumar, "Performance of shared-memory switches under multicast bursty traffic," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 473–488, April 1997.

[65] M. Grossglauser, "Optimal deterministic timeouts for reliable scalable multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 422–433, April 1997.

[66] S. Floyd, V. Jacobson, S. McCanne, C. G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," in *ACM SIGCOMM*, pp. 342–356, August 1995.

[67] A. Erramilli and R. P. Singh, "A reliable and efficient multicast protocol for boadband broadcast networks," in *Proc. of ACM SIGCOMM*, August 1987.

[68] S. J. Golestani and K. Sabnani, "Fundamental observations on multicast congestion control in the Internet," in *Proc. of IEEE INFOCOM*, March 1999.

[69] M. Gerla and L. Kleinrock, "Flow control: a comparative survey," *IEEE Trans. on Communications*, vol. 28, no. 4, pp. 553–574, April 1980.

[70] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. on Communications*, vol. 29, no. 7, pp. 954–962, July 1981.

[71] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, March 1998.

[72] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, January–February 1997.

[73] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multirate multicast sessions," in *Proc. of IEEE INFOCOM*, pp. 123–132, April 2001.

[74] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," in *IEEE Conference on Decision and Control*, December 2001.

[75] J. Widmer and M. Handley, "Extending equation-based congestion control to multicast applications," in *Proc. of ACM SIGCOMM*, 2001.

[76] Michael Luby, Lorenzo Vicisano, and Tony Speakman, *Heterogeneous multicast congestion control based on router packet filtering*, Reliable Multicast Transport (RMT) Working Group, May 1999.

[77] L. Rizzo, "pgmcc: a TCP-friendly single-rate multicast congestion control scheme," in *Proc. of ACM SIGCOMM*, 2000.

[78] R. Jain, D. M. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared systems," *Technical Report, DEC-TR-301, Digital Equipment Corporation*, 1984.

[79] S. Bhattacharyya, D. Towsley, and J. Kurose, "The loss path multiplicity problem in multicast congestion control," in *Proc. of IEEE INFOCOM*, pp. 856–863, March 1999.

[80] D. P. Bertsekas, *Nonlinear promgramming*, Athena Scientific, 2nd edition, 1999.

[81] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation – numerical methods*, Prentice Hall, 1989.

[82] S. Paul, K. K. Sabanani, J. C.-H. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, April 1997.

[83] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. X. Zhang, "A reliable multicast framework for light-weight sessions adn application level framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784–802, December 1997.

[84] S. Paul, K. K. Sabanani, and D. M. Kristol, "Multicast transport protocols for high speed networks," in *Proc. of IEEE International Conference on Network Protocols (ICNP)*, April 1994.

[85] R. Yavatkar, J. Griffioen, and M. Sudan, "A reliable dissemination protocol for interactive collaborative applications," in *Proc. of ACM Multimedia*, November 1995.

[86] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. of ACM SIGCOMM*, pp. 117–130, 1996.

[87] L. Vicisano, L. Rizzo, J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. of IEEE INFOCOM*, pp. 996–1003, March 1998.

[88] S. Kasera, J. Kurose, and D. Towsley, "Scalable reliable multicast using multiple multicast groups," in *Proc. of ACM SIGMETRICS*, pp. 64–74, June 1997.

[89] S. K. Kasera, G. Hjalmtysson, D. Towsley, and J. Kurose, "Scalable reliable multicast using multiple channels," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 294–310, June 2000.

[90] A. Achary and B. R. Badrinath, "Delivering multicast messages in networks with mobile hosts," in *Proc. 13th International Conference on Distributed Computing Systems*, May 1993.

[91] A. Achary, A. Bakre, and B. R. Badrinath, "IP multicast extension for mobile networking," in *Rutgers DCS Technical Report LCSR-TR-243*, May 1995.

[92] H. Wang and M. Schwartz, "Performance analysis of multicast flow control algorithms over combined wired/wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1349–1361, September 1997.

[93] H. Wang and M. Schwartz, "Performance analysis of multicast flow control algorithms over combined wired/wireless networks," in *Proc. of IEEE INFOCOM*, April 1997.

[94] D. Duchamp, "Issues in wireless in mobile computing," in *Third Workshop on Workstation Operating Systems*, July 1992.

[95] K. Cho and K. Birman, "A group communication approach for mobile computing," presented at *Workshop on Mobile Computing Systems and Applications*, July 1994.

[96] P. Bhagwat and C. E. Perkins, "A mobile networking system based on Internet protocol (IP)," in *Proc. USENIX Symp. Mobile and Location Independent Computing*, August 1993.

[97] R. Caceres and L. Iftode, "The effect of mobility on reliable transport protocols," in *Proc. 14th International Conference on Distributed Computer Systems*, 1994.

[98] M. S. Corson and S. G. Batsell, "A reservation based multicast (RBM) routing protocol for mobile networks: overview of initial route construction," in *Proc. of IEEE INFOCOM*, pp. 1063–1074, April 1995.

[99] S. Dolev, D. K. Pradhan, and J. L. Welch, "Modified tree structure for location management in mobile environments," in *Proc. of IEEE INFOCOM*, pp. 530–537, April 1995.

[100] J. Ioannidis and G. Q. Maguire, "The design and implementation of a mobile internetworking architecture," in *Proc. USENIX Conference*, 1993.

[101] D. B. Johnson, "Scalable and robust internetwork routing for mobile hosts," in *Proc. 14th International Conference on Distributed Computer Systems*, June 1994.

[102] S. H. Brae, S.-J. Lee, W. Su, and M. Gerla, "Evaluation of the on-demand multicast routing protocol in multihop wireless," *IEEE Network Magazine*, vol. 14, no. 1, pp. 70–77, January/February 2000.

[103] K. Sabnani and M. Schwartz, "Multidestination protocols for satellite broadcast channels," *IEEE Trans. on Communications*, vol. 13, , 1985.

[104] M. H. Ammar and L. R. Wu, "Improving the throughput of point-to-multi-point ARQ protocols through destination set splitting," in *Proc. of IEEE INFOCOM*, pp. 262–271, May 1992.

[105] X. Li, M. Ammar, and S. Paul, "Video multicast over the internet," *IEEE Network Magazine*, March 1999.

[106] D. G. Luenberger, *Optimization by Vector Space Method*, John Wiley and Sons, 1969.

[107] D. G. Luenberger, *Linear and nonlinear programming*, Addison-Wesley Publishing Company, 1984.